JUMBO BLOCK Rescue
Benutzerhandbuch

JUMBO BLOCK Rescue

Resilienz endet nicht am Bauwerk.

Wer JUMBO BLOCK bisher vor allem mit Regenwasserrückhaltung, Wasserspeicherung und dem Schutz gefährdeter Infrastruktur vor Starkregen und Überflutung verbindet, kennt nur einen Teil dessen, was hinter JUMBO BLOCK steht.

Denn kritische Infrastruktur wird nicht allein durch Beton geschützt.

Sie funktioniert nur, wenn **Bauwerk, Wasserwirtschaft, Technik, Daten, Dokumentation, Kommunikation und IT** auch unter außergewöhnlichen Bedingungen zusammenspielen.

Genau hier setzt **JUMBO BLOCK Rescue** an.

JUMBO BLOCK Rescue ist ein eigenständiges, bootfähiges Rescue-, Diagnose-, Engineering- und Arbeitssystem für Situationen, in denen die reguläre IT-Infrastruktur nicht mehr oder nur noch eingeschränkt zur Verfügung steht.

Es wurde nicht entwickelt, um ein weiteres Betriebssystem zu sein.

Es wurde entwickelt, damit man **weiterarbeiten kann, wenn das eigentliche System es nicht mehr kann.**

KRITIS bedeutet: mit Störungen rechnen, bevor sie eintreten

Mit dem KRITIS-Dachgesetz rückt die Widerstandsfähigkeit kritischer Anlagen noch stärker in den Mittelpunkt.

Für Betreiber bedeutet Resilienz deshalb mehr als die Frage, wie eine Anlage im Normalbetrieb funktioniert.

Die entscheidenden Fragen beginnen dort, wo der Normalbetrieb endet:

Was passiert bei Starkregen oder Überflutung?

Was passiert bei einem beschädigten Datenträger?

Was passiert, wenn ein Rechner nicht mehr startet?

Was passiert nach einem Cybervorfall?

Sind Projektunterlagen, technische Dokumentationen und Engineering-Werkzeuge noch verfügbar, wenn zentrale IT-Systeme ausfallen?

Kann eine Organisation ihre technischen Informationen noch lesen, prüfen, sichern und weiterverarbeiten?

Und ganz praktisch:

Kann überhaupt noch gearbeitet werden?

JUMBO BLOCK Rescue ist unsere Antwort auf einen Teil genau dieser Fragen.

Ein USB-System. Vier technische Ebenen.

JUMBO BLOCK Rescue verbindet auf einem bootfähigen System vier Werkzeugwelten, die normalerweise getrennt betrachtet werden.

A – JB-Rescue

Die erste Ebene ist die eigentliche technische Rettungs- und Diagnoseumgebung.

35 spezialisierte Werkzeuge stehen für Datenrettung, Datenträgerdiagnose, Partitionierung, Dateisysteme, Malware- und Rootkit-Prüfung, Netzwerkanalyse, Verschlüsselung, Hardwarediagnose, Remote-Zugriff und technische Dokumentenbearbeitung zur Verfügung.

TestDisk, PhotoRec, GNU ddrescue, GParted, SMART- und NVMe-Diagnose, ClamAV, Wireshark, Nmap, Cryptsetup, GnuPG und viele weitere Werkzeuge bilden eine umfangreiche technische Werkzeugkette.

Dabei gilt ein Grundsatz:

Erst verstehen. Dann sichern. Dann handeln.

Originale und beschädigte Datenträger sollen möglichst unverändert bleiben. Instabile Medien werden zuerst gesichert. Analysen und Reparaturen erfolgen, wo immer möglich, an Images oder Arbeitskopien.

Rescue bedeutet für uns nicht hektisches Reparieren.

Rescue bedeutet **kontrolliertes technisches Vorgehen**.

B – MX-Werkzeuge

Die zweite Ebene ergänzt diese Werkzeugkette um gezielt ausgewählte Funktionen der technischen MX-Basis.

Boot-Reparatur, UEFI-Verwaltung, Chroot-Rettung, Erstellung bootfähiger Rescue-Medien, Systemabbilder und Signaturprüfung unterstützen Wiederherstellung und Betrieb der Umgebung selbst.

MX ist dabei die technische Basis von JUMBO BLOCK Rescue – **nicht dessen Zweck**.

JUMBO BLOCK Rescue ist ausdrücklich dafür gedacht, auch Systeme und Daten anderer Betriebssystemwelten zu untersuchen, zu sichern und wiederherzustellen.

C – Zubehör

Ein Notfallsystem muss mehr können, als nur Fehler zu finden.

Deshalb gehören auch praktische Arbeitswerkzeuge zur Umgebung: Dateimanager, Dateisuche, Archivverwaltung, Texteditor, Hash-Prüfung, Netzwerkverwaltung, Terminal und Screenshot-Werkzeuge.

Denn in einer realen Störung müssen Informationen nicht nur gerettet werden.

Sie müssen auch:

gefunden, geprüft, kopiert, dokumentiert und weitergegeben werden können.

D – JUMBO BLOCK Engineering Suite

Und genau an dieser Stelle geht JUMBO BLOCK Rescue einen entscheidenden Schritt weiter.

Das System enthält zusätzlich die **JUMBO BLOCK Engineering Suite**.

Damit endet die Funktion des Rescue-Systems nicht bei:

„Der Rechner läuft wieder."

Es schafft zusätzlich eine lokale Arbeitsumgebung für wesentliche technische, projektbezogene und dokumentarische Prozesse.

Hydrologie, Anlagenplanung, Budgetierung, Inspektion, Projektmanagement, Ausschreibung, wissenschaftliche Berechnungen, Einheitenumrechnung, Notizen, Tabellen, Dokumenterstellung, Dokumentenanzeige, Text- und Codebearbeitung, PDF-Werkzeuge, Konvertierung, Archivierung und Dateiverwaltung stehen unmittelbar innerhalb der Rescue-Umgebung zur Verfügung.

Die Anwendungen arbeiten weitgehend lokal und sind bewusst darauf ausgelegt, zentrale Cloud-Dienste für ihre Kernfunktionen nicht vorauszusetzen.

Das bedeutet:

Rescue endet nicht bei der Wiederherstellung von Daten.

Es kann der Punkt sein, von dem aus die Arbeit weitergeht.

JUMBO BLOCK ist weit mehr als Regenwasserrückhaltung

Der Ursprung von JUMBO BLOCK liegt in einer sehr physischen Herausforderung:

Wasser dort beherrschbar zu machen, wo es für Infrastruktur zur Gefahr wird.

Unter versiegelten Flächen entstehen Retentions- und Speicherräume. Niederschlagswasser kann zurückgehalten, gespeichert und kontrolliert bewirtschaftet werden, während die Oberfläche weiterhin infrastrukturell genutzt werden kann.

Gerade im Umfeld kritischer Infrastruktur ist Retention aber kein isoliertes Thema.

Starkregen kann:

- Zufahrten überfluten,
- technische Anlagen gefährden,
- Betriebsflächen unbenutzbar machen,
- Energie- und Kommunikationsinfrastruktur beeinträchtigen,
- Betriebsunterbrechungen auslösen.

Wer solche Risiken betrachtet, erkennt schnell:

Physische und digitale Resilienz treffen sich am selben Standort.

Ein überfluteter Technikbereich ist kein reines Wasserproblem.

Ein ausgefallener Rechner ist kein reines IT-Problem.

Ein verlorener Engineering-Datensatz ist kein reines Dateiproblem.

Und ein vorhandener Notfallplan hilft wenig, wenn man im entscheidenden Moment nicht mehr auf ihn zugreifen kann.

Deshalb denken wir JUMBO BLOCK nicht als einzelnes Betonprodukt.

Wir denken es als **Infrastrukturkonzept**.

Von der Retention zur Resilienz

JUMBO BLOCK verbindet deshalb unterschiedliche technische Disziplinen:

Wasserwirtschaft.
Konstruktiver Ingenieurbau.
Infrastrukturplanung.
Engineering.
Inspektion und Dokumentation.
Datenintegrität.
IT-Resilienz.
Recovery.

Das mag auf den ersten Blick ungewöhnlich erscheinen.

Für uns ist es konsequent.

Denn die entscheidende Frage lautet nicht:

„Welches Produkt haben wir?“

Sondern:

„Was muss funktionieren, wenn es darauf ankommt?“

Aus dieser Perspektive gehören ein unterirdischer Retentionsraum, ein statisches Engineering-Modell, eine lokale Projektdatenbank, ein SHA-256-Hash, ein Datenrettungswerkzeug und ein bootfähiges Rescue-System plötzlich zu derselben Idee:

Handlungsfähigkeit erhalten.

Für alle, die KRITIS nicht nur auf dem Papier denken

JUMBO BLOCK Rescue richtet sich an Menschen und Organisationen, die Verantwortung für funktionierende Infrastruktur tragen.

An Betreiber.

An technische Verantwortliche.

An Ingenieurinnen und Ingenieure.

An IT-Abteilungen.

An Kommunen und Versorger.

An Industrie, Rechenzentren und andere Betreiber technisch anspruchsvoller Standorte.

Und an alle, die sich im Zuge steigender Resilienzanforderungen irgendwann eine sehr einfache Frage stellen müssen:

Was haben wir tatsächlich zur Verfügung, wenn unsere gewohnte Infrastruktur ausfällt?

JUMBO BLOCK Rescue ist kein Ersatz für ein vollständiges Notfall-, Informationssicherheits- oder Business-Continuity-Management.

Es ist auch kein Versprechen, dass sich jedes beschädigte System oder jeder verlorene Datensatz wiederherstellen lässt.

Es ist etwas sehr Praktisches:

eine umfangreiche technische Werkzeug- und Arbeitsumgebung, die dann verfügbar sein soll, wenn die normale Umgebung es möglicherweise nicht mehr ist.

70 Werkzeuge und Funktionen. Ein Gedanke.

Die Baseline 20261001 von JUMBO BLOCK Rescue umfasst die in diesem Handbuch dokumentierten vier Blöcke mit insgesamt **70 Programmen und Funktionen**.

Doch die Zahl allein ist nicht das Entscheidende.

Entscheidend ist, wie diese Werkzeuge zusammenspielen.

Ein Datenträger kann diagnostiziert werden.

Ein beschädigtes Medium kann zunächst gesichert werden.

Partitionen und Dateisysteme können untersucht werden.

Verlorene Dateien können gesucht werden.

Netzwerkverkehr kann analysiert werden.

Hardware kann diagnostiziert werden.

Daten können verschlüsselt und ihre Integrität geprüft werden.

Gerettete Dateien können durchsucht, sortiert und archiviert werden.

Technische Dokumente können gelesen und bearbeitet werden.

Projektinformationen können lokal weiterverarbeitet werden.

Und wesentliche Werkzeuge der JUMBO BLOCK Engineering Suite stehen unmittelbar zur Verfügung.

Das ist der Unterschied zwischen einer Sammlung von Hilfsprogrammen und einer **Rescue-Umgebung**.

Vielleicht ist genau das KRITIS.

Nicht die Vorstellung, dass niemals etwas ausfällt.

Sondern die Fähigkeit, vorbereitet zu sein, **wenn etwas ausfällt**.

JUMBO BLOCK schützt Infrastruktur deshalb nicht nur dort, wo Wasser sichtbar wird.

Wir beschäftigen uns mit der Frage, wie technische Systeme widerstandsfähiger werden – **unter der Erde, auf der Fläche, im Engineering und in den Daten.**

JUMBO BLOCK Rescue macht diese Haltung greifbar.

Bootfähig.
Lokal.
Offline nutzbar.
Technisch nachvollziehbar.
Und gemacht für den Moment, in dem Improvisation allein nicht mehr genügt.

JUMBO BLOCK Rescue

Weil Resilienz nicht am Bauwerk endet.

Hinweis zum Inhaltsverzeichnis

Dieses Handbuch besitzt bewusst ein **ausführliches Inhaltsverzeichnis**.

JUMBO BLOCK Rescue vereint eine große Zahl unterschiedlicher Werkzeuge für Datenrettung, Diagnose, Netzwerk, Sicherheit, Systemwiederherstellung, allgemeine Arbeitsaufgaben und die JUMBO BLOCK Engineering Suite. Entsprechend unterschiedlich können auch die Situationen sein, in denen dieses Handbuch benötigt wird.

Das Inhaltsverzeichnis soll deshalb nicht nur einen Überblick über den Aufbau des Handbuchs geben. Es ist zugleich als **schneller technischer Wegweiser** gedacht.

Wer JUMBO BLOCK Rescue zum ersten Mal verwendet, kann sich anhand der vier Blöcke einen strukturierten Überblick über das System verschaffen.

Wer bereits weiß, welches Werkzeug benötigt wird, kann gezielt zum entsprechenden Programm oder Themenbereich springen.

Und im konkreten Rescue-Fall soll möglichst schnell die entscheidende Frage beantwortet werden:

„Wo finde ich das Werkzeug und die Informationen, die ich jetzt brauche?“

Deshalb werden die einzelnen Programme und Funktionen im Inhaltsverzeichnis nicht künstlich zusammengefasst oder verborgen. Die detaillierte Gliederung ist beabsichtigt.

Inhaltsverzeichnis

JUMBO BLOCK Rescue....2
Resilienz endet nicht am Bauwerk....2
KRITIS bedeutet: mit Störungen rechnen, bevor sie eintreten....2
Ein USB-System. Vier technische Ebenen....3
A – JB-Rescue....3
B – MX-Werkzeuge....3
C – Zubehör....3
D – JUMBO BLOCK Engineering Suite....4
JUMBO BLOCK ist weit mehr als Regenwasserrückhaltung....4
Von der Retention zur Resilienz....5
Für alle, die KRITIS nicht nur auf dem Papier denken....6
70 Werkzeuge und Funktionen. Ein Gedanke....6
Vielleicht ist genau das KRITIS....7
Hinweis zum Inhaltsverzeichnis....8
Block A....65
Werkzeuggruppen....65
Zuordnung der Bereiche, Programme und Versionen....67
A.1 TestDisk....69
Zweck....69
Typische Einsatzfälle....69
Start und grundlegende Bedienung....69
Analyse einer Partitionstabelle....70

Wiederherstellung einer Partition....70
Wiederherstellung gelöschter Dateien....70
Bootsektoren und Dateisystemstrukturen....71
Vorgehen bei physisch beschädigten Datenträgern....71
TestDisk und PhotoRec....71
Sicherheitshinweise....72
Rolle in JUMBO BLOCK Rescue....72
A.2 PhotoRec....73
Zweck....73
Typische Einsatzfälle....73
Arbeitsprinzip....73
Start und Auswahl des Datenträgers....74
Auswahl der Quelldaten....74
Auswahl des Dateisystemtyps....74
Gesamte Partition oder nur freier Speicher....74
Auswahl der wiederherzustellenden Dateitypen....75
Auswahl des Zielverzeichnisses....75
Durchführung der Wiederherstellung....75
Dateinamen und Verzeichnisstruktur....76
Optionen für schwierige Fälle....76
Arbeiten mit Datenträgerabbildern....77
Fragmentierte Dateien....77
Fortsetzen einer unterbrochenen Wiederherstellung....77
Sicherheitshinweise....77
TestDisk oder PhotoRec?....78
Rolle in JUMBO BLOCK Rescue....78
A.3 GNU ddrescue....79
Zweck....79
Abgrenzung zu dd und dd_rescue....79
Typische Einsatzfälle....79
Grundprinzip....80
Quelle und Ziel....80
Das Mapfile....80
Fortsetzen eines Rettungsvorgangs....81
Rettungsphasen....81
Erster Rettungsdurchlauf....81
Weitere Rettungsversuche....82
Direkter Datenträgerzugriff....82
Schreiben auf ein Blockgerät....82
Image-Datei oder physischer Klon....83
Zielmedium richtig dimensionieren....83
Dateisystemunabhängiges Arbeiten....83
Datenträger möglichst nicht einhängen....84
Keine Reparatur durch ddrescue....84
Umgang mit einem ausfallenden Datenträger....84
Grenzen....85
Sicherheitshinweise....85
Rolle in JUMBO BLOCK Rescue....85
A.4 Foremost....87
Zweck....87
Typische Einsatzfälle....87

Arbeitsprinzip....87
Unterstützte integrierte Dateitypen....88
Grundlegender Aufruf....88
Auswahl des Ausgabeordners....89
Auswahl bestimmter Dateitypen....89
Verbose Mode....90
Quick Mode....90
Quiet Mode....90
Audit-only-Modus....91
Option -a....91
Blockgröße....91
Überspringen von Bereichen....91
Indirect Block Detection....92
Zeitgestempelte Ausgabeordner....92
Eigene Dateisignaturen....92
Aufbau von foremost.conf....93
Binäre Signaturen....93
Audit-Datei....93
Foremost und Datenträgerabbilder....93
Grenzen des File Carving....94
Ursprüngliche Dateinamen und Verzeichnisstruktur....94
Foremost und PhotoRec....94
Sicherheitshinweise....95
Rolle in JUMBO BLOCK Rescue....95
A.5 Extundelete....96
Zweck....96
Einsatzbereich....96
Grundlegender Aufruf....96
Wiederherstellung einer einzelnen Datei....97
Wiederherstellung eines Verzeichnisses....97
Wiederherstellung aller auffindbaren Dateien....97
Mehrere ausgewählte Dateien wiederherstellen....98
Wiederherstellung anhand einer Inode-Nummer....98
Superblock untersuchen....98
Journal untersuchen....99
Inode-Informationen anzeigen....99
Blockinhalt anzeigen....99
Zeitliche Eingrenzung....99
Externes Journal....100
Backup-Superblock....100
Warum schnelles Handeln wichtig ist....101
Original oder Abbild....101
Keine Garantie auf Wiederherstellung....101
Abgrenzung zu TestDisk....102
Abgrenzung zu PhotoRec und Foremost....102
Abgrenzung zu GNU ddrescue....102
Grenzen....102
Sicherheitshinweise....103
Rolle in JUMBO BLOCK Rescue....103
A.6 GParted....104
Zweck....104

Bedeutung im Rescue-System....104
Programmoberfläche....104
Partitionen untersuchen....105
Unterstützte Operationen....105
Partitionstabelle erstellen....105
Neue Partition erstellen....106
Partition löschen....106
Partition formatieren....106
Größe einer Partition ändern....106
Partition verschieben....107
Partition kopieren und einfügen....107
Partition prüfen....107
Partitionsmarkierungen....107
Ausstehende Operationen....108
Vor dem Anwenden kontrollieren....108
Eingehängte Partitionen....108
Verschlüsselte Partitionen....109
Dateisystemunterstützung....109
Datenträgertypen....109
Besonderheit von GParted 1.6.0....110
GParted und TestDisk....110
GParted und GNU ddrescue....110
Typischer Rescue-Ablauf....110
Grenzen....111
Sicherheitshinweise....111
Rolle in JUMBO BLOCK Rescue....111
A.7 Laufwerke / GNOME Disks....113
Zweck....113
Datenträgerübersicht....113
Datenträger und Blockgeräte....114
Partitionen und Volumes....114
Einhängen und Aushängen....114
SMART-Daten....114
SMART-Selbsttests....115
Abgrenzung zu A.8 – SMART Diagnose....115
Leistungstest....115
Schreibbenchmark....116
Datenträgerabbilder....116
Datenträgerabbild erstellen....116
Datenträgerabbild wiederherstellen....117
Abgrenzung zu GNU ddrescue....117
Partitionen erstellen und verwalten....117
Formatieren....118
Formatieren ist kein sicheres Löschen....118
Dateisystemgröße verändern....118
Dateisystem prüfen und reparieren....118
Laufwerk ausschalten....119
GNOME Disks und GParted....119
Typischer Rescue-Ablauf....119
Grenzen....120
Sicherheitshinweise....120

Rolle in JUMBO BLOCK Rescue....120
A.8 SMART Diagnose....122
Zweck....122
Was ist SMART?....122
Unterstützte Geräteklassen....122
Gerät identifizieren....123
Geräte automatisch suchen....123
Geräteinformationen anzeigen....123
SMART-Gesundheitsstatus....124
Umfangreiche SMART-Informationen....124
Erweiterte Gesamtausgabe....125
SMART-Attribute bei ATA-Geräten....125
Normalisierte Werte und Rohwerte....126
Fehlerprotokolle....126
Selbsttestprotokoll....126
Interne SMART-Selbsttests....126
Kurzer Selbsttest....127
Langer beziehungsweise erweiterter Selbsttest....127
Selbsttest abbrechen....127
Captive-Modus....127
SMART aktivieren oder deaktivieren....128
Gerätetyp vorgeben....128
USB-Gehäuse und Adapter....128
RAID-Controller....129
NVMe und SMART....129
Temperatur....129
SMART ist kein Oberflächenscan....129
SMART ist kein Datenrettungsprogramm....130
SMART Diagnose und GNU ddrescue....130
SMART Diagnose und GNOME Disks....130
JSON-Ausgabe....130
Laufwerksdatenbank....131
Grenzen der SMART-Diagnose....131
Sicherheitshinweise....131
Typischer Rescue-Ablauf....132
Rolle in JUMBO BLOCK Rescue....132
A.9 NVMe Diagnose....134
Zweck....134
NVMe....134
Controller und Namespace....135
Erkannte NVMe-Geräte auflisten....135
Controller identifizieren....135
Namespace identifizieren....136
SMART-/Health-Informationen....136
Globale und Namespace-bezogene SMART-Daten....136
Typische Health-Informationen....137
Critical Warning....137
Temperatur....138
Available Spare....138
Percentage Used....138
Data Units Read und Data Units Written....138

Power Cycles und Power On Hours....138
Unsafe Shutdowns....139
Media and Data Integrity Errors....139
Fehlerprotokoll....139
Anzahl der Fehlerprotokolleinträge....139
SMART-Log im JSON-Format....140
Rohdaten ausgeben....140
Device Self-Test....140
Kurzer Selbsttest....141
Erweiterter Selbsttest....141
Status des Selbsttests....141
Selbsttest abbrechen....141
Auf Selbsttest warten....142
Selbsttestprotokoll....142
Abgrenzung zur SMART Diagnose....142
Diagnose versus Administration....143
Format....143
Namespace-Verwaltung....143
Schreibbefehle....144
Firmware-Funktionen....144
Controller Reset....144
Sichere erste Diagnose....144
Verhalten bei auffälligen NVMe-Laufwerken....145
NVMe Diagnose und GNU ddrescue....145
Grenzen....145
Sicherheitshinweise....146
Versionshinweis....146
Rolle in JUMBO BLOCK Rescue....147
A.10 ClamAV Scan....148
Zweck....148
Signaturdatenbank....148
Grundsyntax....148
Einzelne Datei scannen....149
Verzeichnis scannen....149
Rekursiver Scan....149
Gesamtes System scannen....150
Scan eines eingehängten Datenträgers....150
Fundmeldungen....150
Scan-Zusammenfassung....150
Scanbericht in Datei schreiben....151
Nur infizierte Dateien anzeigen....151
Quiet-Modus....151
Akustische Fundmeldung....151
Alternative Signaturdatenbank....152
Nur offizielle Signaturen....152
PUA-Erkennung....152
Archive und Container....153
Maximale Dateigröße....153
Maximale Scanmenge pro Container....153
Dateiliste verwenden....153
Dateisystemgrenzen....153

Infizierte Dateien verschieben....154
Infizierte Dateien kopieren....154
Dateien automatisch löschen....155
Heuristische Warnungen....155
Verschlüsselte Archive und Dokumente....155
Temporäre Dateien....156
clamscan und clamdscan....156
Kein vollständiges Endpoint-Security-System....156
Scan eines Windows-Datenträgers....156
Vorteil eines extern gestarteten Rescue-Systems....157
Fund ist nicht gleich Beweis....157
Zugriffsrechte....157
Scan und beschädigte Datenträger....157
ClamAV und Datenrettung....158
Typischer Rescue-Ablauf....158
Sicherheitshinweise....159
Rolle in JUMBO BLOCK Rescue....159
A.11 ClamAV Signaturen....161
Zweck....161
Bedeutung der Signaturdatenbanken....161
FreshClam....162
Aktualisierungsablauf....162
Inkrementelle Aktualisierung....162
Beschädigte Datenbank....162
Datenbanktypen....163
CVD....163
CDIFF....163
Digitale Signaturen....163
Manuelle Aktualisierung....164
Interaktiver Betrieb....164
Daemon-Modus....164
DNS-basierte Versionsprüfung....164
Proxy-Unterstützung....165
Konfigurationsdatei....165
Konfiguration prüfen....165
Datenbankverzeichnis....165
Datenbankbesitz und Rechte....166
Update-Protokoll....166
Netzwerkverbindung....166
Offline-Betrieb....167
Offline bedeutet nicht wirkungslos....167
Aktualisierung vor einem Scan....167
Keine Internetverbindung erzwingen....167
Kontrolliertes Offline-Update....168
FreshClam und lokale Mirrors....168
Signaturen und Engine-Kompatibilität....168
Fehler beim Update....169
Wiederholte Updateversuche....169
Signaturen sind keine Garantie....169
Signaturstand und Scanbericht....170
Trennung von Update und Scan....170

Typischer Rescue-Ablauf mit Internetverbindung....170
Typischer Rescue-Ablauf ohne Internetverbindung....170
Sicherheitshinweise....171
Rolle in JUMBO BLOCK Rescue....171
A.12 Chkrootkit....172
Zweck....172
Was ist ein Rootkit?....172
Bestandteile von Chkrootkit....172
Grundaufruf....173
Ausgaben....173
Verfügbare Tests anzeigen....174
Einzelnen Test ausführen....174
Quiet-Modus....174
Expert-Modus....174
Debug-Modus....175
Optionen nicht zusammenziehen....175
Externes System untersuchen....175
Vorteil der externen Untersuchung....176
Alternative vertrauenswürdige Programme verwenden....176
Bedeutung für JUMBO BLOCK Rescue....177
NFS-Verzeichnisse überspringen....177
Dateisystemtypen ausschließen....177
Bekannte False Positives ausschließen....177
Sniffer-Test und False Positives....178
Sniffer-Ausnahmen....178
Promiscuous Mode....178
Prüfung von Systemprogrammen....179
LKM-Prüfungen....179
Login-Spuren....180
chkwtmp....180
chklastlog....180
Bekannte Rootkit-Merkmale....180
Chkrootkit ist kein Virenscanner....181
Keine automatische Bereinigung....181
False Positives....181
Kein Fund ist ebenfalls kein Beweis....182
Vertrauensproblem eines kompromittierten Systems....182
Typischer Rescue-Ablauf....182
Vorgehen bei einer Warnung....182
Beweissicherung....183
Chkrootkit auf einem beschädigten Datenträger....183
Chkrootkit und ClamAV....183
Grenzen....183
Sicherheitshinweise....184
Rolle in JUMBO BLOCK Rescue....184
A.13 Rkhunter....186
Zweck....186
Grundlegendes Prüfprinzip....186
Grundlegender Systemcheck....187
Interaktiver Ablauf....187
Warnungen....187

Kein Fund ist keine Sicherheitsgarantie....188
Unterstützte Prüfungen anzeigen....188
Einzelne Prüfungen aktivieren....188
Prüfungen deaktivieren....189
Konfiguration....189
Konfiguration überprüfen....189
Dateieigenschaften....190
Hashverfahren....190
Properties-Datenbank....190
--propupd....191
--propupd nicht blind verwenden....191
Selektives --propupd....191
Paketmanager-Unterstützung....192
Rkhunter-Daten aktualisieren....192
Unterschied zwischen --update und Softwareupdate....192
Versionsprüfung....192
Offline-Betrieb....193
Kein Netzwerkzwang....193
Protokollierung....193
Vorheriges Protokoll....194
Nur Warnungen anzeigen....194
Zusammenfassung....194
Farben....195
Quiet-Modus....195
Netzwerkprüfungen....195
Systemstartdateien....195
Versteckte Dateien und Verzeichnisse....196
Rootkit-spezifische Prüfungen....196
Shared-Memory-Prüfung....196
Zusätzliche Hilfsprogramme....196
Chkrootkit und Rkhunter....196
Rkhunter und ClamAV....197
Besonderheit eines Rescue-Systems....197
Konsequenz für JUMBO BLOCK Rescue....197
Warnungen richtig behandeln....197
--propupd niemals als Warnungsbeseitigung....198
Physisch beschädigte Datenträger....198
Keine automatische Bereinigung....198
Grenzen....198
Sicherheitshinweise....199
Typischer Rescue-Ablauf....199
Rolle in JUMBO BLOCK Rescue....199
A.14 GUFW Firewall....201
Zweck....201
Einsatzgebiet im Rescue-System....201
GUFW und UFW....202
Firewall aktivieren und deaktivieren....202
Status prüfen....202
Grundprinzip einer hostbasierten Firewall....203
Eingehender und ausgehender Verkehr....203
Standardrichtlinien....203

Typischer Sicherheitsansatz....204
Regeln....204
Allow....204
Deny....205
Reject....205
Limit....205
Ports....205
Protokolle....206
IP-Adressen....206
Subnetze....206
Regelreihenfolge....206
Regeln löschen....207
Profile....207
Logging....207
GUFW und Logging....208
Firewall und Netzwerkdiagnose....208
GUFW und SSH....208
GUFW und Nmap....208
GUFW und Wireshark....209
GUFW und ClamAV....209
Reset....209
Firewall deaktivieren ist keine Diagnose....209
Firewall schützt nicht vor allen Angriffen....210
Ausgehender Verkehr....210
Unbekannte Netzwerke....211
Rescue-System als Wartungsrechner....211
Firewall-Regeln vor Änderungen dokumentieren....211
Typischer Rescue-Ablauf....212
Sicherheitshinweise....212
Grenzen....212
Technische Einordnung in JUMBO BLOCK Rescue....213
A.15 Binwalk....214
Zweck....214
Typische Einsatzgebiete....214
Grundlegender Aufruf....215
Offsets....215
Signaturanalyse....215
Signaturfund ist kein vollständiger Beweis....216
Ausgabe ungültiger Treffer....216
Bestimmte Signaturen einschließen....216
Signaturen ausschließen....217
Suche nach einer Bytefolge....217
Opcode-Scan....217
Eigene Magic-Datei....218
Automatische Extraktion....218
Externe Extraktionsprogramme....219
Extraktionsverzeichnis....219
Rekursive Extraktion....219
Rekursionstiefe....220
Anzahl extrahierter Dateien begrenzen....220
Größe extrahierter Dateien begrenzen....220

Carving ohne Extraktionsprogramme....221
Rekursive Analyse mit Vorsicht....221
Entropieanalyse....221
Bedeutung hoher Entropie....222
Schnelle Entropieanalyse....222
Entropiediagramm speichern....222
Scanbereich begrenzen....222
Basisadresse....223
Byte-Reihenfolge verändern....223
Ergebnisse protokollieren....224
CSV-Ausgabe....224
Quiet-Modus....224
Verbose-Modus....224
Typisches erstes Vorgehen....225
Beispiel einer Firmware-Untersuchung....225
Binwalk und Hexedit....225
Binwalk und Foremost....226
Binwalk und PhotoRec....226
Binwalk ist kein Malware-Scanner....226
Binwalk ist kein Disassembler....226
Binwalk ist kein Datenrettungsprogramm für beschädigte Hardware....226
Originaldateien nicht unnötig verändern....227
Nicht vertrauenswürdige Extraktionsinhalte....227
Externe Hilfsprogramme....227
False Positives....227
Grenzen....228
Sicherheitshinweise....228
Technische Einordnung in JUMBO BLOCK Rescue....228
A.16 Hexedit....230
Zweck....230
Einsatzgebiet im Rescue-System....230
Grundlegender Aufruf....231
Bildschirmaufbau....231
Hexadezimale Werte....231
Offset....232
Navigation....232
Zum Dateianfang springen....232
Zum Dateiende springen....232
Zu einem bestimmten Offset springen....233
Dezimale und hexadezimale Positionen....233
Suche....233
Suche nach Zeichenketten....233
Suche nach Hexadezimalwerten....234
Suche fortsetzen....234
Direkte Bearbeitung....234
Hexadezimale Bearbeitung....234
Zeichenbasierte Bearbeitung....235
Umschalten zwischen Hex- und Zeichenbereich....235
Änderungen sind zunächst editierbar....235
Änderungen speichern....236
Beenden....236

Read-only-Modus....236
Read-only als Standard für Untersuchungen....236
Geräte und Datenträger....237
Direkter Schreibzugriff auf Blockgeräte....237
Image statt Original....237
Physisch beschädigte Datenträger....238
Binwalk und Hexedit....238
Beispiel eines Analyseablaufs....238
Hexedit und Dateisignaturen....239
Beschädigte Datei-Header....239
Keine automatische Interpretation....239
Keine automatische Strukturprüfung....239
Überschreiben statt strukturelles Editieren....240
Zeichenkodierung....240
Suche ist keine Dateianalyse....240
Keine Malware-Erkennung....240
Kein Disassembler....241
Kein File-Carving....241
Keine automatische Datenrettung....241
Typische Rescue-Szenarien....241
Sicherer Arbeitsablauf....242
Forensische Betrachtung....242
Grenzen....242
Sicherheitshinweise....242
Technische Einordnung in JUMBO BLOCK Rescue....243
A.17 Nmap....244
Zweck....244
Einsatzgebiet im Rescue-System....244
Rechtliche und betriebliche Voraussetzung....245
Grundsyntax....245
Standardscan....245
Host Discovery....246
Nur Host Discovery....246
„Ping Scan“ bedeutet nicht nur ICMP....246
Lokales Ethernet....246
Host Discovery überspringen....247
Konsequenz von -Pn....247
Ziele auflisten ohne Scan....247
Ports....247
Standardmäßig 1.000 Ports....248
Bestimmte Ports auswählen....248
Alle Ports....248
Fast Scan....248
Top Ports....249
TCP und UDP unterscheiden....249
Die sechs Portzustände....249
open....250
closed....250
filtered....250
unfiltered....250
open|filtered....251

closed|filtered....251
Ergebnisse sind perspektivabhängig....251
TCP SYN Scan....251
SYN Scan und Rechte....252
TCP Connect Scan....252
Unterschied zwischen -sS und -sT....252
UDP Scan....252
UDP-Ergebnisse....252
UDP-Scans können langsam sein....253
TCP und UDP kombinieren....253
ACK Scan....253
Weitere Scanverfahren....254
Service Detection....254
Versions- und Diensterkennung....254
Mögliche -sV-Informationen....255
nmap-service-probes....255
Intensität der Versionserkennung....255
Light Version Scan....256
Alle Version-Probes....256
Betriebssystemerkennung....256
OS-Fingerprint....256
OS-Erkennung ist eine Schätzung....257
Aggressivere OS-Schätzung....257
Aggressive Detection....257
-A bewusst einsetzen....258
Nmap Scripting Engine....258
Standardskripte....258
NSE ist nicht grundsätzlich passiv oder harmlos....259
NSE-Kategorien....259
safe bedeutet nicht risikofrei....259
Intrusive Scripts....260
vuln....260
exploit, brute und dos....260
Einzelnes NSE-Skript....260
Hilfe zu einem NSE-Skript....261
Skriptargumente....261
Ausgabe....261
Normale Ausgabe....261
XML-Ausgabe....261
Grepable Output....262
Mehrere Formate gleichzeitig....262
Verbose-Ausgabe....262
Nur offene beziehungsweise potenziell offene Ports....263
Netzwerkinterfaces und Routen....263
Timing....263
Timing Templates nicht mit Scanqualität verwechseln....263
Umfang beeinflusst die Laufzeit....263
Firewall-Einfluss....264
GUFW und Nmap....264
Nmap und ARP Scan....264
Nmap und Wireshark....265

Nmap und TCPDump....265
Nmap und MTR....265
Nmap und Dig....265
Typischer Rescue-Ablauf....265
Beispiel: Unbekanntes Gerät im lokalen Netz....266
Beispiel: Dienst nicht erreichbar....266
Beispiel: UDP-Dienst....267
Scanergebnisse dokumentieren....267
Grenzen....267
Sicherheitshinweise....267
Technische Einordnung in JUMBO BLOCK Rescue....268
A.18 Wireshark....269
Zweck....269
Einsatzgebiet im Rescue-System....269
Wireshark ist kein Portscanner....270
Live Capture und Offline-Analyse....270
Capture Interfaces....270
Mehrere Interfaces....271
Capture-Rechte....271
Dumpcap....271
Promiscuous Mode....271
Grenzen in geswitchten Netzen....272
WLAN und Monitor Mode....272
Start einer Aufzeichnung....272
Die drei zentralen Ansichtsbereiche....273
Packet List....273
Packet Details....273
Packet Bytes....274
Protokoll-Dissektoren....274
Portnummer ist nicht gleich Protokoll....274
Decode As....274
Capture Filter und Display Filter....275
Capture Filter....275
Capture-Filter-Syntax....275
Capture Filter sparsam einsetzen....275
Display Filter....276
Beispiel tcp....276
Beispiel dns....276
Beispiel IP-Adresse....276
Beispiel TCP-Port....276
Quell- und Zielrichtung....277
Logische Verknüpfungen....277
Filter verändern die Daten nicht....277
Paket suchen....277
Pakete markieren....277
Zeitdarstellung....278
Zeitstempel....278
TCP-Verbindungsaufbau....278
TCP-Verbindungsabbau....279
TCP Retransmissions....279
Checksums....279

DNS-Analyse....279
DHCP-Analyse....280
ARP-Analyse....280
ICMP-Analyse....280
HTTP....281
TLS....281
Verschlüsselung bleibt Verschlüsselung....281
Follow Stream....281
Follow TCP Stream....282
Follow Stream verändert Capture nicht....282
Statistics....282
Protocol Hierarchy....282
Endpoints....282
Conversations....283
I/O Graphs....283
Expert Information....283
Expert Information ist kein endgültiges Urteil....283
Capture-Dateien....284
PCAPNG....284
PCAP....284
Capture-Dateien enthalten sensible Daten....284
Capture-Dateien nicht unkontrolliert weitergeben....285
Capture File Properties....285
Capture-Kommentare....285
Ring Buffer....286
Automatisches Stoppen....286
Paketverluste beim Capture....286
Snap Length....286
Namensauflösung....287
Export von Paketen....287
Export von Packet Bytes....287
Export Objects....287
Capinfos....287
Editcap....288
Mergecap....288
Wireshark und TCPDump....288
Wireshark und Nmap....288
Wireshark und GUFW....289
Wireshark und Dig....289
Wireshark und MTR....289
Typischer Rescue-Ablauf....289
Beispiel: DNS-Problem....290
Beispiel: TCP-Dienst nicht erreichbar....291
Beispiel: DHCP-Problem....291
Grenzen....291
Sicherheitshinweise....292
Technische Einordnung in JUMBO BLOCK Rescue....292
A.19 TCPDump....294
Zweck....294
Einsatzgebiet im Rescue-System....294
Grundlegender Aufruf....294

Netzwerkinterfaces anzeigen....295
Interface auswählen....295
any....295
Capture-Berechtigungen....295
Standardausgabe....296
Paketanzahl begrenzen....296
Namensauflösung deaktivieren....296
Ports ebenfalls numerisch darstellen....297
Verbose-Ausgabe....297
Zeitstempel....297
Absolute Zeit....298
Filterausdruck....298
BPF-Filter....298
Host filtern....298
Quellhost....299
Zielhost....299
Netzwerk filtern....299
Port filtern....299
TCP filtern....299
UDP filtern....300
TCP-Port....300
UDP-Port....300
DNS-Verkehr....300
ARP....300
ICMP....301
IPv6....301
Logische Verknüpfungen....301
AND....301
OR....301
NOT....302
Klammern....302
Capture Filter reduzieren die Datenmenge....302
Snaplen....302
Verkürzte Pakete....303
Capture in Datei schreiben....303
-w speichert Pakete, nicht Terminaltext....303
Capture-Datei lesen....303
Filter auf gespeicherte Captures anwenden....304
Offline-Analyse....304
Wireshark-Übergabe....304
Terminal und GUI ergänzen sich....304
ASCII-Ausgabe....305
Hex- und ASCII-Ausgabe....305
Hex-Ausgabe....305
Link-Layer-Header....305
Paketinhalt kann sensibel sein....306
Richtung des Captures....306
Eingehender Verkehr....306
Ausgehender Verkehr....307
Promiscuous Mode....307
Promiscuous Mode sieht nicht automatisch das gesamte LAN....307

Capture-Puffer....307
Paketverluste....307
Ein unvollständiger Capture kann täuschen....308
Rotation nach Dateigröße....308
Rotation nach Zeit....308
Anzahl der Dateien begrenzen....308
Langzeit-Captures....309
Sofortige Ausgabe....309
Capture über Pipe....309
Keine Paketinjektion....309
TCPDump ist überwiegend passiv....309
TCPDump und Nmap....310
TCPDump und Wireshark....310
TCPDump und ARP Scan....310
TCPDump und MTR....311
TCPDump und Dig....311
TCPDump und GUFW....311
Typischer Rescue-Ablauf....311
Beispiel: DNS funktioniert nicht....312
Beispiel: TCP-Verbindung....312
Beispiel: ARP-Problem....313
Beispiel: Capture für Wireshark....313
Beispiel: Nur begrenzte Paketmenge....313
Capture-Dateien als sensible Daten....313
Datenschutz und Berechtigung....314
Verschlüsselter Verkehr....314
Grenzen....314
Sicherheitshinweise....315
Technische Einordnung in JUMBO BLOCK Rescue....315
A.20 ARP Scan....317
Zweck....317
Einsatzgebiet im Rescue-System....317
ARP....318
Grundprinzip....318
ARP arbeitet nur lokal....318
Konsequenz für den Rescue-Einsatz....318
IPv4....318
Grundsyntax....319
Einzelnes Ziel....319
Mehrere Ziele....319
CIDR-Netz....319
Adressbereich....319
Netzwerk und Netzmaske....320
Lokales Netz automatisch bestimmen....320
Bedeutung von --localnet....320
Netzwerk- und Broadcastadresse....320
Interface auswählen....321
Automatische Interface-Auswahl....321
Interface ohne IPv4-Adresse....321
Berechtigungen....321
Standardausgabe....322

MAC-Adresse....322
MAC-Adressen können geändert werden....322
Vendor-Erkennung....322
IEEE-Zuordnungen....323
Zusätzliche Zuordnungen....323
Herstellerangabe richtig interpretieren....323
Virtualisierung....323
Keine Vendorinformation....323
OUI-Datei aktualisieren....324
Minimalistische Ausgabe....324
Nur Hostzeilen....324
Benutzerdefiniertes Ausgabeformat....324
Hostnamen auflösen....325
Nur numerische Zieladressen....325
Ziele aus Datei....325
--file und --localnet....326
Wiederholungsversuche....326
Warum Wiederholungen?....326
Weniger Wiederholungen....326
Timeout....326
Backoff....327
Bandbreite....327
Warum Bandbreite begrenzen?....327
Höhere Bandbreite....327
Paketintervall....328
Duplikate....328
Doppelte Antworten können relevant sein....328
ARP und Firewalls....328
Ein nicht pingbares Gerät kann sichtbar sein....329
Ein ARP-Fund beweist keinen Dienst....329
ARP Scan und Nmap....329
ARP Scan und Wireshark....329
ARP Scan und TCPDump....330
ARP Scan und MTR....330
Typischer Rescue-Ablauf....330
Beispiel: IP-Adresse eines Geräts unbekannt....331
Beispiel: Gerät antwortet nicht auf Ping....331
Beispiel: Unbekanntes Gerät....331
Beispiel: Falsches Interface....332
Grenzen....332
Sicherheitshinweise....332
Technische Einordnung in JUMBO BLOCK Rescue....332
A.21 MTR....334
Zweck....334
Einsatzgebiet im Rescue-System....334
Grundprinzip....335
Schrittweise TTL....335
Unterschied zu Traceroute....335
Unterschied zu Ping....336
Grundlegender Aufruf....336
Interaktive Anzeige....336

Loss%....337
Snt....337
Last....337
Avg....337
Best....337
Wrst....337
StDev....337
Latenz....337
Jitter und Schwankungen....338
Der wichtigste Interpretationsfehler....338
Warum?....338
Entscheidendes Interpretationsprinzip....339
Fortgesetzter Verlust....339
Ziel ist besonders wichtig....339
Nicht antwortende Hops....339
Firewall und Filter....339
Namensauflösung....340
DNS deaktivieren....340
Hostname und IP-Adresse gemeinsam....340
IPv4 erzwingen....341
IPv6 erzwingen....341
Warum IPv4 und IPv6 getrennt testen?....341
Report Mode....341
Anzahl der Messzyklen....342
Bericht in Datei speichern....342
Warum mehrere Messzyklen?....342
Wide Report....342
Beispiel für Rescue-Dokumentation....343
Intervall....343
Messlast....343
Paketgröße....343
Warum Paketgröße relevant sein kann....344
Interface auswählen....344
Quelladresse festlegen....344
Report-Felder anpassen....345
JSON....345
XML....345
CSV....346
ICMP....346
UDP....346
TCP....346
Zielport....346
Warum verschiedene Probe-Protokolle wichtig sind....347
Beispiel....347
MPLS....347
Autonomous-System-Informationen....347
Interaktive Steuerung....348
Statistik zurücksetzen....348
Route kann sich ändern....348
Load Balancing....348
Asymmetrisches Routing....349

ICMP-Priorisierung....349
Beispiel: scheinbarer Verlust an einem Zwischen-Hop....349
Beispiel: Verlust setzt sich fort....349
Beispiel: Latenzsprung....350
Geografische Distanz....350
Beispiel: starke Schwankungen....350
Typischer Rescue-Ablauf....350
MTR und Nmap....351
MTR und Wireshark....352
MTR und TCPDump....352
MTR und ARP Scan....352
MTR und Dig....352
Grenzen....352
Sicherheitshinweise....353
Technische Einordnung in JUMBO BLOCK Rescue....353
A.22 Dig....354
Zweck....354
Einsatzgebiet im Rescue-System....355
DNS ist ein eigenes Diagnosesystem....355
Grundsyntax....356
Standardresolver....356
Kein verwendbarer Resolver eingetragen....356
Bestimmten DNS-Server abfragen....356
Warum @server wichtig ist....357
Direkte Serverabfrage und Systemresolver unterscheiden....357
Aufbau der Standardausgabe....357
Header....358
Status....358
NOERROR....358
NXDOMAIN....358
Keine Antwort vom Server....359
QUESTION SECTION....359
ANSWER SECTION....359
TTL....360
AUTHORITY SECTION....360
ADDITIONAL SECTION....360
Query Time....360
SERVER....360
MSG SIZE....361
A Record....361
AAAA Record....361
A und AAAA getrennt untersuchen....361
CNAME....361
MX....362
NS....362
SOA....362
Seriennummer....363
TXT....363
CAA....363
SRV....363
PTR und Reverse DNS....364

IPv4 Reverse DNS....364
IPv6 Reverse DNS....364
Reverse DNS ist nicht Forward DNS....364
Kurzausgabe....365
Wann +short sinnvoll ist....365
Nur Answer Section....365
IPv4-Transport erzwingen....365
IPv6-Transport erzwingen....366
Wichtige Unterscheidung....366
UDP und TCP....366
Warum TCP testen?....367
DNS verwendet nicht nur UDP....367
DNS over TLS....367
Nichtstandard-Port....367
Timeout....368
Wiederholungsversuche....368
Rekursion....368
Rekursiver Resolver....369
Autoritativer Nameserver....369
Cache-Unterschiede....369
+trace....369
Was +trace zeigt....369
+trace ist keine normale Resolverabfrage....370
@server zusammen mit +trace....370
DNSSEC-Daten....370
+dnssec bedeutet nicht automatisch vollständige Validierung....370
+trace und DNSSEC....370
DNS-Flags....371
aa....371
ra....371
ad....371
Mehrere Abfragen....371
Batch-Datei....372
Benutzerkonfiguration .digrc....372
Reproduzierbare Diagnose....372
Exitcodes....372
Bedeutung für Skripte....373
Zone Transfer....373
AXFR ist kein allgemeines Discovery-Verfahren....373
Typischer Rescue-Ablauf....373
Beispiel: Browser meldet „Server nicht gefunden“....374
Beispiel: Dig erhält keine Antwort....374
Beispiel: A funktioniert, AAAA nicht....375
Beispiel: Resolver liefert alte Daten....375
Beispiel: Delegationsproblem....375
Beispiel: UDP/TCP-Unterschied....375
Dig und Wireshark....376
Dig und TCPDump....376
Dig und MTR....376
Dig und Nmap....376
Dig und ARP Scan....377

Grenzen....377
Sicherheitshinweise....377
Technische Einordnung in JUMBO BLOCK Rescue....378
A.23 Whois....380
Zweck....380
Einsatzgebiet im Rescue-System....380
WHOIS ist kein DNS....381
WHOIS ist keine Geo-IP-Datenbank....381
WHOIS ist keine Personensuche....381
Grundsyntax....382
Automatische Serverauswahl....382
Domain abfragen....382
IP-Adresse abfragen....383
IPv6-Adresse....383
AS-Nummer....383
Autonomous System....383
MTR und Whois....383
Bestimmten WHOIS-Server auswählen....384
Warum einen Server ausdrücklich auswählen?....384
Port festlegen....384
WHOIS verwendet TCP....384
IANA zuerst abfragen....385
Datenschutz bei -I....385
Referral....385
Registry und Registrar....385
Thin Registry....386
Referral-Rekursion deaktivieren....386
Warum Referral-Kontrolle wichtig ist....386
Ausführliche Programmausgabe....386
Rechtliche Hinweise ausblenden....386
Registry-Hinweise können wichtig sein....387
RIPE-artige WHOIS-Dienste....387
Wichtige Konsequenz....387
Kurze IP-/Abuse-Ausgabe....387
Objektfilterung deaktivieren....387
Exakte Netzpräfixsuche....388
Weniger spezifische Objekte....388
Spezifischere Objekte....388
Warum Netz-Hierarchien relevant sind....388
Quellen auswählen....388
Serverinformationen....389
Objekttypen....389
Objektvorlagen....389
Inverse Suche....389
Kontakt-Rekursion deaktivieren....390
Referral auf RIPE-artigen Servern....390
Konfigurationsdatei....390
Aufbau von whois.conf....390
Bedeutung für Rescue-Diagnosen....391
Umgebungsvariable WHOIS_SERVER....391
Umgebungsvariable WHOIS_OPTIONS....391

Internationalisierte Domains....391
ACE / Punycode....391
Zeichenkodierung....392
WHOIS-Daten sind nicht einheitlich....392
Automatische Auswertung ist daher schwierig....392
Domain-WHOIS und Datenschutz....392
Registrant ist nicht zwingend Betreiber....393
IP-WHOIS und Netzzuordnung....393
IP-Zuordnung beweist keine Benutzeridentität....393
ASN-Zuordnung....394
Registrierung und tatsächliches Routing....394
Whois und Dig....394
Whois und MTR....394
Whois und Nmap....395
Whois und Wireshark/TCPDump....395
Whois und ARP Scan....395
Typischer Rescue-Ablauf: unbekannte öffentliche IP-Adresse....395
Typischer Rescue-Ablauf: unbekannte Domain....396
Typischer Rescue-Ablauf: MTR-Hop einordnen....396
Typischer Rescue-Ablauf: AS-Nummer....397
Private IPv4-Adressen....397
Loopback und reservierte Adressen....397
Offline-Betrieb....397
WHOIS-Server kann nicht erreichbar sein....397
Rate Limits....398
Datenschutz und Nutzungsbedingungen....398
Daten können veraltet sein....398
WHOIS und RDAP....398
Grenzen....399
Sicherheitshinweise....399
Technische Einordnung in JUMBO BLOCK Rescue....399
A.24 Chromium....401
Zweck....401
Einsatzgebiet im Rescue-System....402
Browser und Netzwerkdiagnose....402
Typische Diagnosekette....403
Direkte IP-Adresse....403
Bedeutung für die Diagnose....404
Hostname kann technisch erforderlich sein....404
HTTP....404
HTTPS....405
Zertifikate....405
Zertifikatsfehler....405
Lokale Appliances und selbstsignierte Zertifikate....405
Vor dem Umgehen einer Zertifikatswarnung....406
Systemzeit....406
Sicherheitsanzeige....406
Weiterleitungen....406
HTTP-Statuscodes....407
Beispiele....407
Developer Tools....408

DevTools öffnen....408
Network Panel....408
Was das Network Panel sichtbar macht....408
Diagnosebeispiel....409
Request Headers....409
Response Headers....410
Payload....410
Response....410
Initiator....411
Timing....411
Preserve Log....411
Cache deaktivieren....411
Hard Reload....411
Filter....412
Fetch/XHR....412
WebSocket....412
HAR....413
HAR-Dateien können hochsensibel sein....413
DevTools ist kein vollständiger Paketmitschnitt....413
Wichtige Trennung....414
Inkognito-Modus....414
Sinn im Rescue-System....414
Inkognito bedeutet nicht Anonymität....414
Downloads im Inkognito-Modus....415
Lesezeichen im Inkognito-Modus....415
Inkognito und forensische Anforderungen....415
Downloads....415
Download bedeutet nicht Vertrauenswürdigkeit....415
Downloads verifizieren....416
Download aus unbekannter Quelle....416
ClamAV als Ergänzung....416
GnuPG als Ergänzung....416
Sandbox....417
Sandbox ist keine absolute Sicherheitsgrenze....417
chromium-sandbox....417
Site Isolation....417
Defense in Depth....417
Kein Grund, Sicherheitsmechanismen abzuschalten....418
Lokale Geräte mit alten Weboberflächen....418
Sicherheitsgrenzen nicht für Kompatibilität opfern....418
Captive Portals....418
Proxy....418
Cookies....419
Cookies können sicherheitsrelevant sein....419
Local Storage und andere Site-Daten....419
Cache....419
JavaScript....420
Console....420
Vorsicht mit Console-Befehlen....420
Passwort- und Zugangsdaten....420
Rescue-System als temporäre Arbeitsumgebung....421

Typischer Rescue-Ablauf: lokale Routeroberfläche nicht erreichbar....421
Typischer Rescue-Ablauf: Hostname funktioniert nicht....422
Typischer Rescue-Ablauf: Webseite lädt teilweise....422
Typischer Rescue-Ablauf: HTTPS-Warnung bei lokaler Appliance....422
Typischer Rescue-Ablauf: Download einer Reparaturdatei....423
Chromium und Dig....423
Chromium und MTR....423
Chromium und Nmap....424
Chromium und TCPDump....424
Chromium und Wireshark....424
Chromium und Whois....424
Grenzen....425
Sicherheitshinweise....425
Technische Einordnung in JUMBO BLOCK Rescue....426
A.25 Cryptsetup....429
Zweck....429
Einsatzgebiet im Rescue-System....429
Cryptsetup und dm-crypt....430
Wichtige Ebenentrennung....430
LUKS....431
LUKS1 und LUKS2....431
LUKS erkennen....431
Ausführliche Prüfung....431
Bedeutung von isLuks....432
Keine LUKS-Erkennung....432
Header untersuchen....432
luksDump....432
luksDump entschlüsselt nicht die Nutzdaten....433
UUID....433
UUID nicht mit Dateisystem-UUID verwechseln....433
Keyslots....433
Passphrase ist nicht direkt der Datenverschlüsselungsschlüssel....433
Konsequenz....434
LUKS-Volume öffnen....434
Beispielprinzip....434
Mapping-Name....434
Passphrase....435
Falsche Passphrase....435
Read-only öffnen....435
Warum Read-only wichtig ist....435
Read-only allein genügt nicht immer für einen vollständigen Schutz....435
Geöffnetes Mapping prüfen....436
Status eines nicht aktiven Mappings....436
Nach dem Öffnen....436
Dateisystem untersuchen....436
Verschlüsselung und Dateisystemschaden unterscheiden....436
LUKS erfolgreich öffnen bedeutet nicht „Daten gesund“....437
Mapping schließen....437
Vor dem Schließen....437
Warum die Reihenfolge wichtig ist....438
Rescue-Grundablauf....438

Beschädigter Datenträger....438
Verschlüsseltes Image....439
Vorteil....439
Verschlüsselung verhindert kein Imaging....439
Header als kritische Struktur....439
LUKS1....439
LUKS2....439
Header-Backup....440
Bedeutung des Header-Backups....440
Header-Backup enthält Schlüsselzugangsdaten....440
Konsequenz....440
Header-Backup verändert die Löschstrategie....440
Header wiederherstellen....441
Höchste Vorsicht bei luksHeaderRestore....441
Rescue-Regel....441
Header testen, ohne ihn sofort zurückzuschreiben....441
Detached Header....441
Bedeutung für isLuks....442
Keyfile....442
Keyfile nicht mit Header-Backup verwechseln....442
Schlüssel hinzufügen....442
Schlüssel entfernen....442
Keyslot gezielt löschen....443
Passphrase ändern....443
Rescue-Grundsatz für Keyslots....443
luksFormat....443
Niemals zur „Reparatur“ verwenden....443
Besonders gefährlicher Denkfehler....443
erase / luksErase....444
Kein normales Datenlöschen....444
convert....444
config....444
Tokens....444
Plain dm-crypt....444
Konsequenz bei Plain dm-crypt....445
Keine sichere automatische Erkennung von Verschlüsselung allein anhand „zufälliger Daten“....445
LUKS erleichtert die Verwaltung....445
Keine Passwortwiederherstellung....445
Verschlüsselung und Datenrettung....445
Beispiel 1: LUKS intakt, Dateisystem beschädigt....446
Beispiel 2: LUKS-Header beschädigt....446
Beispiel 3: Passphrase unbekannt....446
Beispiel 4: Physisch defektes verschlüsseltes Laufwerk....446
Beispiel 5: Volume lässt sich öffnen....447
Cryptsetup und GNU ddrescue....447
Cryptsetup und SMART/NVMe Diagnose....447
Cryptsetup und GParted....447
Cryptsetup und GNOME Disks....447
Cryptsetup und GnuPG....447
Typischer sicherer Rescue-Workflow....448

Grenzen....448
Sicherheitshinweise....449
Technische Einordnung in JUMBO BLOCK Rescue....449
A.26 GnuPG....451
Zweck....451
Einsatzgebiet im Rescue-System....451
Cryptsetup und GnuPG unterscheiden....452
Digitale Signatur....452
Was eine erfolgreiche Signaturprüfung technisch aussagt....453
Signaturprüfung ist daher zweistufig....453
Detached Signature....453
Detached Signature prüfen....453
Warum beide Dateien ausdrücklich angeben?....454
Beispiel für eine ISO-Prüfung....454
„Good signature“....454
Fingerprint....454
Warum der Fingerprint wichtig ist....455
Vertrauenswürdiger Vergleich....455
Zentrale Rescue-Regel....455
Öffentliche Schlüssel anzeigen....455
Geheime Schlüssel anzeigen....456
Schlüssel importieren....456
Import ist kein Vertrauensbeweis....456
Schlüssel vor dem Import untersuchen....456
Öffentlichen Schlüssel exportieren....456
--armor....457
Binär und ASCII Armor....457
Datei signieren....457
Detached Signature erzeugen....457
ASCII-armored Detached Signature....457
Signaturschlüssel auswählen....458
Cleartext Signature....458
Cleartext Signatures und Binärdateien....458
Verifikation verändert die Datei nicht....458
GnuPG und Prüfsummen....459
Prüfsumme ohne authentisierte Quelle....459
Herstellerdownloads....459
Bereits vertrauenswürdiges GnuPG verwenden....459
Bedeutung für JUMBO BLOCK Rescue....459
Datei für einen Empfänger verschlüsseln....460
Prinzip der Public-Key-Verschlüsselung....460
Vor dem Verschlüsseln Empfängerschlüssel prüfen....460
Entschlüsseln....460
Ausgabe in Datei....461
Signierte verschlüsselte Datei....461
Symmetrische Verschlüsselung....461
Symmetrische Verschlüsselung im Rescue-Einsatz....461
Passphrase....461
Passphrase nicht in Kommandozeile schreiben....462
gpg-agent....462
Passphrase-Caching....462

Bedeutung im Rescue-System....462
Agent beenden....462
Private Schlüssel....462
Private Schlüssel nicht unnötig importieren....463
Public-Key-Verifikation ist read-only bezüglich des geprüften Artefakts....463
gpgv....463
Unterschied gpg und gpgv....463
Vertrauensmodell von gpgv....464
Konsequenz....464
gpgv mit Keyring....464
Detached Signature mit gpgv....464
Rückgabewert von gpgv....464
gpg und Exitcode....464
Signaturprüfung in Skripten....465
Schlüssel-ID versus Fingerprint....465
User IDs....465
Vertrauen....465
„WARNING: This key is not certified with a trusted signature“....466
Rescue-Workflow ohne aufgebautes Web of Trust....466
Schlüsselserver....466
Schlüsselserver lösen das Authentizitätsproblem nicht....466
Netzwerk und Offline-Betrieb....466
Offline-Verifikation....467
Beispiel: Rescue-ISO prüfen....467
Beispiel: Signatur ist ungültig....467
Beispiel: Public Key fehlt....468
Beispiel: Good Signature, unbekannter Schlüssel....468
Beispiel: Download mit Chromium....468
GnuPG und ClamAV....469
GnuPG und Hashwerte....469
GnuPG und Cryptsetup....469
GnuPG und SSH....469
Metadaten bleiben relevant....470
Sichere Löschung....470
Rescue-System und Schlüsselmaterial....470
Grenzen....470
Sicherheitshinweise....471
Technische Einordnung in JUMBO BLOCK Rescue....471
A.27 SSH / OpenSSH....474
Zweck....474
Baseline-Abgrenzung....474
Grundsyntax....475
Benutzername....475
IP-Adresse statt Hostname....475
DNS und SSH....475
Standardport....475
Anderen Port verwenden....475
SSH besteht aus mehreren Sicherheitsstufen....476
Serverauthentisierung....476
Benutzer-Authentisierung....476
Host Key und Benutzer-Key sind unterschiedliche Schlüssel....476

Erste Verbindung....477
Fingerprint prüfen....477
Warum das wichtig ist....477
known_hosts....477
Wiederholte Verbindung....477
Host Key hat sich geändert....478
Legitime Ursachen einer Host-Key-Änderung....478
Sicherheitsrelevante Ursache....478
Gespeicherten Host suchen....478
Alten Host-Key-Eintrag entfernen....478
Sicherheitsregel für ssh-keygen -R....478
StrictHostKeyChecking....479
Host-Key-Prüfung nicht deaktivieren....479
Public-Key-Authentisierung....479
Privater Schlüssel....479
Öffentlicher Schlüssel....479
Identity-Datei auswählen....480
Passphrase für private Schlüssel....480
Schlüssel ohne Passphrase....480
ssh-agent....480
ssh-add....480
Agentzustand beachten....481
Agent Forwarding....481
Agent Forwarding ist sicherheitskritisch....481
Rescue-Regel....481
Jump Host statt Agent Forwarding....481
Vorteil....482
Remote-Shell....482
Sehr wichtige Rescue-Regel....482
Remote-Kommando....482
Beispielprinzip....482
Vorteil....482
Vorsicht mit Remote-Kommandos....483
Pseudo-Terminal....483
IPv4 erzwingen....483
IPv6 erzwingen....483
Verbose-Modus....483
Was Verbose-Ausgabe hilft zu unterscheiden....484
Debugausgabe kann sensible Informationen enthalten....484
Effektive SSH-Konfiguration anzeigen....484
Warum ssh -G hilfreich ist....484
Benutzerkonfiguration....485
Beispielprinzip....485
Vorteil....485
Systemweite Konfiguration....485
Dateiübertragung mit scp....485
Moderne scp-Übertragung....485
Lokale Datei auf Remote-System kopieren....486
Remote-Datei lokal sichern....486
Verzeichnisse....486
scp verwendet SSH-Sicherheit....486

Port bei scp....486
sftp....486
SFTP ist nicht FTP über TLS....486
Rescue-Einsatz für Dateiübertragung....487
Vor dem Upload auf ein beschädigtes System....487
Port Forwarding....487
Local Forwarding....487
Beispielanwendung....488
Remote Forwarding....488
Sicherheitsrisiko bei Forwardings....488
Local Forwarding bindet standardmäßig lokal....488
Dynamisches Forwarding....488
-N....489
SSH-Tunnel ist kein VPN-Ersatz für jede Situation....489
SSH und Nmap....489
SSH und Dig....489
SSH und MTR....489
SSH und GUFW....489
SSH und TCPDump/Wireshark....489
SSH und Chromium....490
SSH und SSHFS....490
Möglicherweise kompromittiertes Zielsystem....490
Keine privaten Schlüssel auf das Ziel kopieren....490
Kein Agent Forwarding zum kompromittierten Ziel....490
Remote-Ausgaben sind nicht automatisch vertrauenswürdig....491
Wichtige Unterscheidung....491
Typischer Rescue-Ablauf: Server erreichbar, SSH funktioniert nicht....491
Typischer Rescue-Ablauf: neuer Server....492
Typischer Rescue-Ablauf: Host-Key-Warnung....492
Typischer Rescue-Ablauf: Datei vom Server sichern....492
Typischer Rescue-Ablauf: internes Webinterface....492
Typischer Rescue-Ablauf: kompromittierungsverdächtiger Server....492
Grenzen....493
Sicherheitshinweise....493
Technische Einordnung in JUMBO BLOCK Rescue....494
A.28 SSHFS....496
Zweck....496
Einsatzgebiet im Rescue-System....496
SSHFS baut auf SSH auf....496
SSHFS ist kein eigener Serverdienst....497
SFTP statt Remote-Shell....497
Grundsyntax....497
Beispiel....497
Lokaler Mountpunkt....497
Benutzername weglassen....498
Remote-Verzeichnis weglassen....498
Gesamtes Root-Dateisystem....498
Remote-Berechtigungen bleiben maßgebend....498
Wichtige Besonderheit lokaler Rechte....499
default_permissions....499
Standardmäßig nur für den mountenden Benutzer....499

allow_other....499
Grundregel....499
Host-Key-Prüfung....499
Host-Key-Warnung....499
SSH-Optionen verwenden....500
Abweichender SSH-Port....500
Bestimmten privaten Schlüssel verwenden....500
Read-only-Mount....500
Bedeutung für JUMBO BLOCK Rescue....501
Read-only schützt nur diesen Zugriffspfad....501
Kein forensischer Snapshot....501
Live-System bleibt live....501
Konsequenz für Beweissicherung....502
Dateien kopieren....502
Wichtige Perspektive....502
Schreiben wirkt remote....502
Rescue-Grundregel....502
Grafische Dateimanager....502
Gefahr durch grafische Programme....503
Programme auf Remote-Dateien anwenden....503
Netzwerk bleibt Teil des Dateizugriffs....503
Performance....503
Verbindung unterbrochen....504
Ursache....504
ServerAliveInterval....504
Beispiel....504
reconnect....504
Kombination....505
Wichtige Einschränkung von reconnect....505
Kein Schutz vor Datenverlust während eines Abbruchs....505
Bedeutung für Rescue-Kopien....505
Hashprüfung....505
delay_connect....505
sshfs_sync....505
no_readahead....506
sync_readdir....506
Caching....506
SSHFS ist kein Cluster-Dateisystem....506
Remote-Datei wird während des Lesens geändert....506
Datenbankdateien....507
Unmount....507
Baseline und fusermount3....507
Beispiel....507
Vor dem Unmount....507
„Device or resource busy“....507
Verbindungsabbruch und blockierte Prozesse....507
Konsequenz....508
Read-only Rescue-Workflow....508
Workflow bei instabiler Netzwerkverbindung....508
Workflow: Logs eines Remote-Servers sichern....509
Workflow: Remote-Sicherungsziel....509

Sensible Rescue-Daten....509
Transportverschlüsselung ist keine Speicherverschlüsselung....509
Kombination mit GnuPG....509
SSHFS und SCP....510
Wann SCP einfacher ist....510
Wann SSHFS sinnvoll ist....510
SSHFS und SFTP....510
SSHFS und Midnight Commander....511
SSHFS und ClamAV....511
SSHFS und Hashberechnung....511
SSHFS und Netzwerkdiagnose....511
Diagnosegrundsatz....512
IPv6-Adresse....512
/etc/fstab....512
Rescue-Einordnung von /etc/fstab....512
Version 3.7.3....512
Versionsgrenze....513
Grenzen....513
Sicherheitshinweise....513
Technische Einordnung in JUMBO BLOCK Rescue....514
A.29 HWInfo....516
Zweck....516
Einsatzgebiet im Rescue-System....516
Hardware-Erkennung statt Gesundheitsdiagnose....517
Grundaufruf....517
Umfang der Ausgabe....518
Hardwareklassen....518
Mehrere Klassen kombinieren....518
Kurzansicht....519
Schneller Systemüberblick....519
CPU....519
Arbeitsspeicher....519
Wichtige Abgrenzung beim Arbeitsspeicher....519
Datenträger....520
Einzelnen Datenträger untersuchen....520
Bedeutung im Rescue-Fall....520
Gerätebezeichnungen niemals allein aus einer früheren Sitzung übernehmen....520
Blockgeräte....521
Partitionen....521
Storage-Controller....521
SCSI....521
RAID-Geräte....522
Bedeutung....522
PCI-Geräte....522
USB-Geräte....522
USB-Controller....523
Netzwerkhardware....523
Netzwerk und Netcard....523
WLAN....523
Hardware erkannt, Netzwerk funktioniert trotzdem nicht....523
Grafikkarte....524

Monitor....524
BIOS....524
Sound....524
Kamera....525
Bluetooth....525
Tastatur und Maus....525
Drucker und Scanner....525
--all....525
--reallyall....525
Ausgabe protokollieren....526
Nutzen eines Hardwarelogs....526
Datenschutz des Logs....526
Konfiguration speichern....526
Gespeicherte Konfiguration anzeigen....527
Gerätezuordnung vergleichen....527
Bedeutung für ein Live-Rescue-System....527
Debug-Ausgabe....527
--verbose....528
Hardwaredatenbank....528
Datenbank ausgeben....528
Version anzeigen....528
Root-Rechte....529
Root nicht pauschal einsetzen....529
Erkennung hängt vom laufenden Kernel ab....529
„Nicht von HWInfo erkannt“ bedeutet nicht zwingend „nicht vorhanden“....529
Mögliche Ursachen fehlender Erkennung....529
HWInfo verändert keine Partitionen....530
HWInfo repariert keine Datenträger....530
Typischer Rescue-Fall: unbekannter Rechner....530
Typischer Rescue-Fall: Festplatte fehlt....530
Typischer Rescue-Fall: USB-Laufwerk wird nicht erkannt....531
Typischer Rescue-Fall: Netzwerk funktioniert nicht....531
Typischer Rescue-Fall: falsche Bildschirmauflösung....531
Typischer Rescue-Fall: Supportbericht....532
HWInfo und Inxi....532
HWInfo und LSHW....532
HWInfo und dmidecode....532
HWInfo und lm-sensors....532
HWInfo und SMART....532
HWInfo und NVMe Diagnose....533
Hardwareerkennung vor Reparatur....533
Keine Gerätebezeichnung raten....533
Besonders kritisch bei ddrescue....533
Hardwarebericht vor Ausbau....533
Live-System als Vorteil....534
Live-System als Grenze....534
Grenzen....534
Sicherheitshinweise....534
Technische Einordnung in JUMBO BLOCK Rescue....535
A.30 Inxi....537
Zweck....537

Einsatzgebiet im Rescue-System....537
Unterschied zu HWInfo....538
Grundaufruf....538
Basisbericht....538
Bedeutung für Rescue....539
Erweiterter Bericht....539
Detailstufen mit -v....539
-v 0....540
-v 1....540
-v 2....540
Höhere Detailstufen....540
Rescue-Regel....541
CPU....541
CPU-Flags....541
CPU-Erkennung ist kein CPU-Test....541
Maschine....541
DMI-Abgrenzung....542
Arbeitsspeicher....542
RAM-Erkennung ist kein RAM-Test....542
Grafik....542
Rescue-Einsatz Grafik....543
Audio....543
Netzwerkgeräte....543
Erweiterte Netzwerkinformationen....543
IP-Adressen....543
Datenschutz....544
-z – sensible Angaben filtern....544
Beispiel....544
Zentrale Support-Regel....544
-Z....545
Datenträger....545
Datenträgerinformationen sind keine vollständige Zustandsdiagnose....545
Partitionen....545
RAID....545
RAID-Ausgabe ist keine RAID-Reparatur....546
USB....546
Bedeutung für Rescue....546
Bluetooth....546
Batterie....546
Rescue-Regel für Notebooks....547
Sensoren....547
Abgrenzung zu A.33....547
Zusätzliche Details mit -x....547
Beispielprinzip....547
Mehr Details können sensible Daten enthalten....548
-a – administrative Details....548
Bedeutung....548
Nicht automatisch -a verwenden....548
Treiberinformationen....548
Hardware erkannt, aber falscher Treiber....549
Kernel....549

Sehr wichtige Abgrenzung....549
Beispiel....549
Dasselbe gilt für Treiber....549
Desktop-Umgebung....550
Betriebssysteminformationen....550
Rescue-Grundregel....550
Ausgabe ohne Farben....550
Nutzen für gespeicherte Reports....550
Ausgabe umleiten....550
Gefilterter Supportbericht....551
Bericht nicht blind veröffentlichen....551
Ausgabe für interne Dokumentation....551
Typischer Rescue-Ablauf: erster Systemüberblick....551
Typischer Rescue-Fall: Netzwerk funktioniert nicht....552
Typischer Rescue-Fall: Grafikproblem....553
Typischer Rescue-Fall: Storage....553
Typischer Rescue-Fall: Hardwarebericht für Support....553
Typischer Rescue-Fall: Notebook....554
Inxi und HWInfo....554
Inxi und LSHW....554
Inxi und dmidecode....554
Inxi und lm-sensors....555
Inxi und SMART/NVMe....555
Externe Hilfsprogramme....555
Root-Rechte....555
Rescue-Regel....556
Grenzen....556
Sicherheitshinweise....556
Technische Einordnung in JUMBO BLOCK Rescue....557
A.31 LSHW....559
Zweck....559
Einsatzgebiet im Rescue-System....559
Unterschied zu HWInfo und Inxi....560
Grundaufruf....560
Root-Rechte....560
Rescue-Regel....560
Hierarchische Darstellung....560
Bedeutung im Rescue-Betrieb....561
Kurzansicht....561
Typischer Einstieg....561
Hardwarepfad....562
Businformationen....562
Bedeutung....562
-short und -businfo....562
Hardwareklassen....563
Klassen ermitteln....563
Mehrere Klassen kombinieren....563
Storage-Diagnose....563
Netzwerkhardware....563
Netzwerkbericht als HTML....564
Konfiguration eines Netzwerkgeräts....564

Hardware erkannt bedeutet nicht Netzwerk funktioniert....564
CPU....564
CPU-Erkennung ist kein Belastungstest....564
Cache....564
Arbeitsspeicher....565
SPD-Erkennung....565
RAM-Inventarisierung ist kein RAM-Test....565
Firmware und Mainboard....565
DMI....565
Abgrenzung zu dmidecode....566
Unterstützte Informationsquellen....566
Linux-Systeminformationen....566
Rescue-Perspektive....566
PCI und USB....566
Numerische Gerätekennungen....567
Nutzen numerischer IDs....567
Beispielprinzip....567
Datenträger....567
Storage-Controller....567
Kombination....567
Datenträger erkannt bedeutet nicht Datenträger gesund....567
Zustandsdiagnose....568
Hardwarebaum und Partitionen....568
HTML-Ausgabe....568
XML-Ausgabe....568
JSON-Ausgabe....568
Ausgabeformat bewusst wählen....569
Bericht in Datei speichern....569
HTML-Bericht....569
JSON-Bericht....569
Sensible Informationen....569
-sanitize....570
Supportbericht....570
Beispielprinzip....570
Filterung trotzdem kontrollieren....570
-notime....570
Nutzen für Vergleichsberichte....571
-quiet....571
Hardwaretests aktivieren oder deaktivieren....571
Dokumentierte Tests....571
Bedeutung von „Test“....572
Beispiel....572
Warum Informationsquellen deaktivieren?....572
SQLite-Dump....572
Bedeutung....572
Versionsabfrage....573
Hilfe....573
Grafische Oberfläche....573
Baseline-Abgrenzung zur GUI....573
Typischer Rescue-Fall: unbekannter Rechner....573
Typischer Rescue-Fall: Datenträger fehlt....574

Typischer Rescue-Fall: Netzwerkadapter funktioniert nicht....574
Typischer Rescue-Fall: USB-Gerät unbekannt....575
Typischer Rescue-Fall: Hardwaredokumentation....575
Typischer Rescue-Fall: maschineller Vergleich....575
HWInfo, Inxi und LSHW gemeinsam....575
Abweichende Ergebnisse....576
Grenzen der Hardwareerkennung....576
Wichtige Schlussfolgerung....576
Hardwaregesundheit....576
Schreibwirkung....577
Trotzdem mit Bedacht einsetzen....577
Live-System-Perspektive....577
Installiertes Zielsystem....577
Sicherheitshinweise....577
Technische Einordnung in JUMBO BLOCK Rescue....578
A.32 DMI Decode....580
Zweck....580
Einsatzgebiet im Rescue-System....580
Entscheidender Unterschied zu HWInfo und LSHW....581
Grundregel....581
Grundaufruf....581
Root-Rechte....582
Datenquelle unter Linux....582
Direkter Speicherzugriff....582
Aufbau eines DMI-Eintrags....582
DMI-Typen....583
Gezielte Abfrage mit --type....583
Unterstützte Typ-Schlüsselwörter....583
BIOS....584
System....584
Mainboard....584
Gehäuse....584
Prozessor....584
Speicher....584
Physical Memory Array....585
Memory Device....585
Leere Speicherbänke....585
Wichtig: maximale Kapazität ist nicht installierter Speicher....585
RAM-Daten sind kein Speichertest....586
Einzelne Werte mit --string....586
Systemhersteller....586
Systemmodell....586
Systemversion....586
Seriennummer....586
System-UUID....586
BIOS-Hersteller....586
BIOS-Version....587
BIOS-Datum....587
Mainboard-Hersteller....587
Mainboard-Modell....587
Mainboard-Seriennummer....587

Gehäuseinformationen....587
Prozessorinformationen....587
Mehrere Prozessoren....587
Verfügbare String-Schlüssel anzeigen....588
Verfügbare Typen anzeigen....588
Bestimmten Handle anzeigen....588
Nutzen der Handles....588
Quiet-Modus....588
Rescue-Einsatz von --quiet....589
Rohdaten als Hexdump....589
Binäre DMI-Sicherung....589
Wichtige Dateiregel....589
Offline-Auswertung....590
Nutzen für Rescue-Dokumentation....590
DMI-Dump ist kein Festplattenabbild....590
OEM Strings....590
OEM-Daten....590
--no-quirks....591
Rescue-Einordnung von --no-quirks....591
--no-sysfs....591
Firmwarequalität ist entscheidend....591
Typische Platzhalter....591
Firmwareangabe und physische Beschriftung können abweichen....592
Seriennummern....592
System-UUID....592
Asset-Tags....592
Datenschutz....592
Besser gezielt abfragen....593
BIOS-/Firmwarediagnose....593
BIOS-Version ist keine Firmwareintegritätsprüfung....593
Mainboarddiagnose....593
Systemidentifikation....593
Datenschutz beachten....593
RAM-Inventarisierung....594
Typischer Rescue-Fall: RAM-Aufrüstung....594
Grund....594
Typischer Rescue-Fall: unbekannter Rechner....594
Typischer Rescue-Fall: BIOS-Update vorbereiten....595
Kritischer Sicherheitshinweis....595
Typischer Rescue-Fall: Hardware vor Austausch dokumentieren....595
Nach Hardwareänderung....595
Typischer Rescue-Fall: Systembericht....596
Typischer Rescue-Fall: nur BIOS-Version benötigt....596
Typischer Rescue-Fall: nur Systemmodell benötigt....596
Typischer Rescue-Fall: Speicherproblem....596
dmidecode und HWInfo....596
dmidecode und Inxi....596
dmidecode und LSHW....597
Widersprüche zwischen Werkzeugen....597
Physische Realität vor Firmwareangabe....597
SMBIOS-Unterstützung von Version 3.6....597

Neue Optionen in 3.6....598
Neuere SMBIOS-Version als dmidecode unterstützt....598
Rescue-Konsequenz....598
Grenzen....598
Sicherheitshinweise....598
Technische Einordnung in JUMBO BLOCK Rescue....599
A.33 Sensoren / lm-sensors....601
Zweck....601
Einsatzgebiet im Rescue-System....601
Zentrale Abgrenzung....602
Drei getrennte Ebenen....602
Grundaufruf....602
Typische Ausgabe....602
Keine feste Sensorliste erwarten....603
Temperaturen....603
Celsius für technische Dokumentation....603
Chipauswahl....603
Adapterinformationen....603
Nutzen....604
Rohdaten....604
Unterschied zur normalen Ausgabe....604
JSON-Ausgabe....604
Nutzen strukturierter Ausgabe....605
Version....605
Hilfe....605
Konfigurationsdateien....605
Bedeutung der Konfiguration....605
Alternative Konfigurationsdatei....606
Konfiguration temporär umgehen....606
Rescue-Bedeutung....606
Sensorgrenzen....606
Grenzwert ist nicht automatisch Herstellerfreigabe....607
Plausibilitätsprüfung....607
Falsche oder unverbundene Sensoreingänge....607
Lüfterdrehzahlen....607
Nullwert beim Lüfter....608
Spannungen....608
Leistungs- und Stromwerte....608
CPU-Sensoren....608
Mehrere CPU-Kerne....609
Sensorwerte sind Momentaufnahmen....609
Wiederholte Beobachtung....609
Keine künstliche Last erforderlich....609
sensors-detect....609
Was sensors-detect untersucht....610
Interaktiver Betrieb....610
Automatischer Betrieb....610
Wichtiger Sicherheitshinweis zu --auto....610
Grundsätzliches Risiko von sensors-detect....610
Herstellerwarnung....611
Konsequenz für JUMBO BLOCK Rescue....611

Empfohlener Rescue-Ablauf....611
Produktionssysteme....611
Rescue-Grundsatz....611
Kernelmodule....611
Live-System-Perspektive....612
Fehlender Sensorwert....612
Fehlender Wert ist kein Beweis für fehlenden Sensor....612
sensors -s....612
Wichtige Abgrenzung....613
Rescue-Regel....613
Konfigurationsanweisungen....613
label....613
compute....613
ignore....613
set....613
Konsequenz....613
Temperatur-Sensortypen....613
Firmwarekonfiguration....614
Hysterese....614
Sensoralarm....614
Alarm bedeutet nicht automatisch Totalschaden....614
--bus-list....614
Busnummern....615
Typischer Rescue-Fall: Rechner schaltet sich unerwartet ab....615
Typischer Rescue-Fall: Lüfter läuft ständig auf hoher Drehzahl....615
Typischer Rescue-Fall: Lüfter steht....616
Typischer Rescue-Fall: ungewöhnlich hohe CPU-Temperatur....616
Typischer Rescue-Fall: offensichtlich falsche Spannung....616
Typischer Rescue-Fall: keine Sensoren gefunden....617
Typischer Rescue-Fall: thermische Kontrolle vor ddrescue....617
Datenträgertemperatur....617
Sensorwert und SMART-Temperatur....617
lm-sensors und HWInfo....618
lm-sensors und Inxi....618
lm-sensors und LSHW....618
lm-sensors und dmidecode....618
Vier unterschiedliche Hardwareperspektiven....618
Schreibende Hilfsprogramme....619
Sicherheitswarnung zu isaset....619
isadump....619
JUMBO BLOCK Rescue Grundsatz....619
Keine automatischen Schreibvorgänge....619
Grenzen....620
Sicherheitshinweise....620
Technische Einordnung in JUMBO BLOCK Rescue....621
A.34 Midnight Commander....623
Zweck....623
Einsatzgebiet im Rescue-System....623
Keine Datenrettungssoftware im engeren Sinn....624
Zwei-Panel-Prinzip....624
Programmstart....624

Start mit einem Verzeichnis....624
Zwei Startverzeichnisse....624
Rescue-Nutzen....625
Navigation....625
Zwischen den Panels wechseln....625
Menü....625
Funktionstasten....625
Rescue-Vorteil....626
Interner Viewer....626
Rescue-Grundsatz....626
Viewer direkt starten....626
Typische Rescue-Anwendungen des Viewers....626
Interner Editor....627
Editor direkt starten....627
Bearbeiten bedeutet Schreiben....627
Besser auf einer Arbeitskopie....627
Kopieren....628
Typischer Rescue-Fall....628
Kopieren ist nicht automatisch Datenrettung....628
Warum?....628
Verschieben....628
Wichtiger Rescue-Unterschied....628
Umbenennen....629
Verzeichnis erstellen....629
Löschen....629
Kritischer Sicherheitshinweis....629
Kein Papierkorb als Rescue-Strategie....629
Mehrere Dateien auswählen....629
Musterbasierte Auswahl....630
Rescue-Vorsicht bei Massenauswahl....630
Dateisuche....630
Inhaltssuche....630
Rescue-Nutzen....630
Suche ist kein File Carving....631
Shell-Kommandozeile....631
Bedeutung für Rescue....631
Vorsicht....631
Subshell....631
Rescue-Bedeutung....632
Archive und virtuelles Dateisystem....632
Archive....632
Nicht mit echtem Mount verwechseln....632
Archiv nicht unnötig verändern....632
Dateitypen und Aktionen....632
Rescue-Regel....632
Ausführbare Dateien....633
Sicherheitsregel bei verdächtigen Dateien....633
Versteckte Dateien....633
Dateirechte und Metadaten....633
Rescue-Bedeutung....633
Kopie ist nicht automatisch forensisch identisch....634

Forensische Sicherung....634
Read-only-Quelle....634
Wichtige Grenze....634
Read-only und Sicherungsziel....634
Geräteidentität zuerst prüfen....635
Kein blindes Vertrauen in Pfadnamen....635
Freier Speicher....635
Interner Editor mcedit....635
Syntaxhervorhebung....635
Konfiguration....636
Rescue-Bedeutung der Benutzerkonfiguration....636
Mausunterstützung....636
Schwarz-Weiß-Darstellung....636
Farbdarstellung....636
Version anzeigen....637
Hilfe....637
Typischer Rescue-Fall: Benutzerdateien sichern....637
Typischer Rescue-Fall: Konfigurationsdatei untersuchen....637
Typischer Rescue-Fall: Konfiguration reparieren....638
Typischer Rescue-Fall: gerettete Dateien sortieren....638
Wichtige Grenze....638
Typischer Rescue-Fall: Logs sichern....638
Typischer Rescue-Fall: Remote-Dateien....639
Sicherheitsgrenze....639
Midnight Commander und SSH....639
Midnight Commander und GNU ddrescue....639
Midnight Commander und TestDisk....640
Midnight Commander und PhotoRec / Foremost....640
Midnight Commander und Extundelete....640
Midnight Commander und GParted....640
Midnight Commander und GNOME Disks....640
Midnight Commander und ClamAV....640
Midnight Commander und Binwalk / Hexedit....641
Grenzen....641
Sicherheitshinweise....641
Technische Einordnung in JUMBO BLOCK Rescue....642
A.35 QPDF....644
Zweck....644
Einsatzgebiet im Rescue-System....644
PDF prüfen....645
Warnungen richtig bewerten....645
Beschädigte PDF neu schreiben....645
Original erhalten....646
Grenzen der Reparatur....646
Seitenzahl prüfen....646
Seiten auswählen und zusammenstellen....646
PDF aufteilen....647
Verschlüsselte PDFs....647
Entschlüsseln....647
Kein Passwortknacker....647
Eingebettete Dateien....647

Sicherheitsregel für Anhänge....648
Digitale Signaturen....648
Typischer Rescue-Fall: PDF lässt sich nicht öffnen....648
Typischer Rescue-Fall: PDF aus PhotoRec oder Foremost....648
Typischer Rescue-Fall: einzelne Seiten retten....649
Typischer Rescue-Fall: verschlüsselte PDF....649
QPDF und andere Rescue-Werkzeuge....649
Grenzen....650
Sicherheitshinweise....650
Technische Einordnung in JUMBO BLOCK Rescue....651
Block B....652
Werkzeuggruppen....652
Zuordnung der Bereiche, Programme und Versionen....653
B.1 MX Boot Repair....654
Zweck....654
Bedeutung für JUMBO BLOCK Rescue....654
GRUB-Konfiguration reparieren....654
GRUB-Bootloader neu installieren....655
UEFI-Systeme....655
Legacy-/MBR-Systeme....655
Initramfs neu erzeugen....655
Typischer Rescue-Ablauf....656
Dual- und Multi-Boot-Systeme....656
Grenzen....656
Sicherheitshinweise....657
Zusammenspiel mit anderen Rescue-Werkzeugen....657
Technische Einordnung in JUMBO BLOCK Rescue....658
B.2 UEFI Manager....659
Zweck....659
Einsatzgebiet im Rescue-System....659
Voraussetzung....659
Vorhandene Boot-Einträge anzeigen....659
Bootreihenfolge ändern....660
Boot Next....660
Einträge aktivieren und deaktivieren....660
Einträge hinzufügen und entfernen....660
Typisches Rescue-Szenario: falsches Betriebssystem startet....660
Typisches Rescue-Szenario: Dual-Boot....661
EFI Stub Installer....661
Frugal EFI Stub Installer....661
Grenzen....661
Sicherheitshinweise....662
Technische Einordnung in JUMBO BLOCK Rescue....662
B.3 Chroot Rescue Scan....663
Zweck....663
Bedeutung für JUMBO BLOCK Rescue....663
Was ist eine chroot-Umgebung?....663
Automatische Suche....664
Mehrere Linux-Installationen....664
Vorbereitung der chroot-Umgebung....664
Shell im Zielsystem....665

Typische Rescue-Aufgaben....665
Netzwerkzugriff....665
Beschädigtes initrd....666
Typischer Rescue-Ablauf....666
Unterschied zu MX Boot Repair....666
Kein zweites gestartetes Betriebssystem....667
Grenzen gegenüber anderen Betriebssystemen....667
Grenzen bei beschädigten Dateisystemen....667
Sicherheitshinweise....667
Technische Einordnung in JUMBO BLOCK Rescue....668
B.4 MX Live USB Maker....669
Zweck....669
Bedeutung für JUMBO BLOCK Rescue....669
Grundlegender Ablauf....669
Full-featured Mode....670
Image Mode....670
Live-System klonen....670
Dry Run....670
Verschlüsseltes Live-USB....670
Erweiterte Optionen....671
Typischer Einsatz....671
Grenzen....671
Sicherheitshinweise....672
Technische Einordnung in JUMBO BLOCK Rescue....672
B.5 MX Snapshot....673
Zweck....673
Einsatzgebiet im Rescue-System....673
Besondere Bedeutung bei kompromittierten Systemen....674
System-Snapshot....674
Personal Snapshot....674
Ausschlüsse....674
Typischer Rescue-Ablauf....675
Keine automatische Dekontamination....675
Weitere Verwendung....675
Grenzen....676
Sicherheitshinweise....676
Technische Einordnung in JUMBO BLOCK Rescue....676
B.6 ISO Signature Verifier....678
Zweck....678
Bedeutung für JUMBO BLOCK Rescue....678
Betriebssystemunabhängige Prüfung....679
Prüfsumme und Signatur....679
Unterschiedliche Signaturverfahren....679
Grundlegender Arbeitsablauf....680
Signaturschlüssel....680
Gültige Signatur bedeutet nicht automatisch vertrauenswürdiger Herausgeber....680
Typischer Rescue-Ablauf....681
Fehlgeschlagene Verifikation....681
Erfolgreiche Verifikation....681
Keine Malware-Prüfung....682
Zusammenspiel mit ClamAV....682

Zusammenspiel mit MX Live USB Maker....682
Eigene Prüfsummen....683
Grenzen....683
Sicherheitshinweise....683
Technische Einordnung in JUMBO BLOCK Rescue....683
Block C....685
Werkzeuggruppen....685
Zuordnung der Bereiche, Programme und Versionen....686
C.1 – Format USB....687
Zweck....687
Einsatzgebiet im Rescue-System....687
Funktionen....687
Bedienung und Arbeitsablauf....688
Wahl des Dateisystems....688
Typische Rescue-Szenarien....689
Bereitstellung eines Datenträgers für gerettete Dateien....689
Übergabe von Untersuchungsergebnissen....689
Datenaustausch zwischen unterschiedlichen Betriebssystemen....689
Grenzen und Sicherheitshinweise....689
Technische Einordnung in JUMBO BLOCK Rescue....690
C.2 – Thunar File Manager....691
Zweck....691
Einsatzgebiet im Rescue-System....691
Funktionen....691
Bedienung und Arbeitsablauf....692
Typische Rescue-Szenarien....692
Übernahme geretteter Dateien....692
Sicherung zugänglicher Benutzerdaten....692
Zusammenstellung von Untersuchungsergebnissen....693
Datenaustausch mit Wechselmedien....693
Grenzen und Sicherheitshinweise....693
Technische Einordnung in JUMBO BLOCK Rescue....693
C.3 – Catfish File Search....695
Zweck....695
Einsatzgebiet im Rescue-System....695
Funktionen....695
Bedienung und Arbeitsablauf....696
Typische Rescue-Szenarien....696
Suche nach Benutzerdokumenten....696
Suche in geretteten Datenbeständen....696
Auffinden von Konfigurations- und Protokolldateien....696
Vorbereitung einer Datenübergabe....696
Grenzen und Sicherheitshinweise....696
Technische Einordnung in JUMBO BLOCK Rescue....697
C.4 – Engrampa Archive Manager....698
Zweck....698
Einsatzgebiet im Rescue-System....698
Funktionen....698
Bedienung und Arbeitsablauf....699
Typische Rescue-Szenarien....699
Öffnen eines geretteten Archivs....699

Extraktion benötigter Dateien....699
Zusammenstellung von Untersuchungsergebnissen....700
Vorbereitung einer Datenübergabe....700
Grenzen und Sicherheitshinweise....700
Technische Einordnung in JUMBO BLOCK Rescue....700
C.5 – FeatherPad....702
Zweck....702
Einsatzgebiet im Rescue-System....702
Funktionen....702
Bedienung und Arbeitsablauf....703
Typische Rescue-Szenarien....703
Prüfung von Konfigurationsdateien....703
Gezielte Konfigurationsänderung....703
Auswertung von Protokolldateien....703
Untersuchung geretteter Textdateien....704
Dokumentation eines Rescue-Einsatzes....704
Grenzen und Sicherheitshinweise....704
Technische Einordnung in JUMBO BLOCK Rescue....704
C.6 – GtkHash....706
Zweck....706
Einsatzgebiet im Rescue-System....706
Funktionen....706
Bedienung und Arbeitsablauf....707
Typische Rescue-Szenarien....707
Kontrolle einer Dateiübertragung....707
Dokumentation geretteter Dateien....707
Prüfung gegen eine veröffentlichte Prüfsumme....708
Prüfung eines geretteten Datenbestandes....708
Grenzen und Sicherheitshinweise....708
Technische Einordnung in JUMBO BLOCK Rescue....708
C.7 – Advanced Network Configuration....710
Zweck....710
Einsatzgebiet im Rescue-System....710
Funktionen....710
Bedienung und Arbeitsablauf....711
Typische Rescue-Szenarien....711
Manuelle Netzwerkkonfiguration....711
Vorbereitung einer SSH-Verbindung....711
Vorbereitung von SSHFS....712
Netzwerkdiagnose....712
Arbeiten in getrennten Netzwerksegmenten....712
Grenzen und Sicherheitshinweise....712
Technische Einordnung in JUMBO BLOCK Rescue....713
C.8 – Xfce Terminal....714
Zweck....714
Einsatzgebiet im Rescue-System....714
Funktionen....714
Bedienung und Arbeitsablauf....715
Typische Rescue-Szenarien....715
Start von Kommandozeilenwerkzeugen....715
Prüfung von Datenträgern und Dateisystemen....716

Remote-Arbeiten....716
Auswertung technischer Informationen....716
Arbeiten mit administrativen Rechten....716
Grenzen und Sicherheitshinweise....716
Technische Einordnung in JUMBO BLOCK Rescue....717
C.9 – Screenshot....718
Zweck....718
Einsatzgebiet im Rescue-System....718
Funktionen....718
Bedienung und Arbeitsablauf....719
Typische Rescue-Szenarien....719
Dokumentation einer Fehlermeldung....719
Dokumentation eines Datenträgerzustands....719
Dokumentation von Einstellungen....720
Dokumentation von Analyseergebnissen....720
Übergabe eines Rescue-Vorgangs....720
Grenzen und Sicherheitshinweise....720
Technische Einordnung in JUMBO BLOCK Rescue....721
Block D....722
Programmgruppen....722
Zuordnung der Bereiche, Programme und Versionen....724
D.1 – Hydro....726
Zweck....726
Rolle innerhalb der Engineering Suite....726
Wesentliche Funktionen....726
Daten und Datenaustausch....727
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....728
Datensicherung....728
Berichte....729
Grenzen und Hinweise....729
Technische Einordnung in JUMBO BLOCK Rescue....729
Weiterführende Dokumentation....730
D.2 – Plan....731
Zweck....731
Rolle innerhalb der Engineering Suite....731
Wesentliche Funktionen....732
Planung über Zielvolumen....732
Planung über Anlagenmaß....732
Manuelle Anpassung....732
Anlagenlayout....732
SABA-Abdichtung....732
Daten und Datenaustausch....733
Datenintegrität....733
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....733
Datensicherung....734
Berichte....734
Grenzen und Hinweise....734
Technische Einordnung in JUMBO BLOCK Rescue....735
Weiterführende Dokumentation....735
D.3 – Budget....736
Zweck....736

Rolle innerhalb der Engineering Suite....736
Wesentliche Funktionen....737
Daten und Datenaustausch....737
Import aus Plan....737
Import aus Static....737
Kalkulationsprinzip....738
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....738
Berichte....738
Grenzen und Hinweise....739
Technische Einordnung in JUMBO BLOCK Rescue....739
Weiterführende Dokumentation....739
D.4 – Static....741
Status innerhalb von JUMBO BLOCK Rescue....741
Zweck des Engineering-Service....741
Ablauf....741
1. Planen....742
2. Projektdaten übergeben....742
3. Ergebnisse erhalten....742
Leistungsumfang des Engineering-Service....742
Abdeckplatte....742
Boden- und Seitenplatten....742
Gesamtsystem....742
Standsicherheit....743
Dauerhaftigkeit und Robustheit....743
Bewehrung und Optimierung....743
Benötigte Projektdaten....743
Ergebnisse und Berichte....744
Statik-Report....744
Produktionsausdruck....744
Kunden-Report....744
Grenzen....744
Bedeutung innerhalb von JUMBO BLOCK Rescue....745
Weiterführende Informationen....745
D.5 – Temp....746
Status innerhalb von JUMBO BLOCK Rescue....746
Zweck des Engineering-Service....746
Ablauf....747
1. Planen....747
2. Projektdaten übergeben....747
3. Ergebnisse erhalten....747
Leistungsumfang des Engineering-Service....747
Wärmespeicher....747
Wärmeabfuhr....747
Wärmeeinträge....747
Medientemperaturen und Betrieb....748
Standortwirkung....748
Standort-TIB....748
Benötigte Projektdaten....748
Ergebnisse und Berichte....749
TIB-Bericht....749
Detailbilanz....749

Klimaprofile....749
Infrastruktur-Szenarien....749
Grenzen....749
Bedeutung innerhalb von JUMBO BLOCK Rescue....750
Weiterführende Informationen....750
D.6 – Inspect....751
Zweck....751
Rolle innerhalb der Engineering Suite....751
Wesentliche Funktionen....752
Grunddaten aus der Projektplanung....752
Anlagenbaum....752
Plan- und Grenzwerte....753
Prüfkatalog....753
Fälligkeiten....754
Inspektionsgründe....754
Inspektion durchführen....754
Fotodokumentation....755
Mängelmanagement....755
Daten und Datenaustausch....755
Eingangsdaten....755
Inspektionsdaten....756
Datenintegrität....756
Datei-Hash....756
Verkettete Inspektionsprotokolle....756
Fotos....757
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....757
Datensicherung....757
Berichte....758
Inspektionsprotokoll....758
Wartungs- und Inspektionsplan....758
Mängelliste....758
Grenzen und Hinweise....758
Technische Einordnung in JUMBO BLOCK Rescue....759
Weiterführende Dokumentation....759
D.7 – Project....760
Zweck....760
Rolle innerhalb der Engineering Suite....760
Wesentliche Funktionen....761
Projekt-Stammdaten....761
Projektabschnitte und Projektschritte....761
Verantwortlicher und Ausführender....761
Status und Termine....762
Termine-Übersicht....762
Automatischer Zeitplan....762
Partner und Rollen....763
Dokumenten-Register und Dateianhänge....763
Statusprotokoll....764
Daten und Datenaustausch....764
Additiver Import und Zusammenführen....765
Konfliktlösung....765
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....765

Datensicherung....766
Berichte....766
Projektbericht....766
Gantt-Bericht....767
Datenintegrität....767
Grenzen und Hinweise....767
Technische Einordnung in JUMBO BLOCK Rescue....768
Weiterführende Dokumentation....768
D.8 – Tender....769
Zweck....769
Rolle innerhalb der Engineering Suite....769
Wesentliche Funktionen....770
GAEB-X-Dateien öffnen....770
Neues Leistungsverzeichnis....770
LV-Struktur....771
Positionen....771
Leitpositionen und Unterpositionen....772
Suche und Navigation....772
Preise und Summen....773
Datenphasen und Phasenwechsel....773
X80....773
X82....773
X82 → X83....773
X84....774
X84-Angebotsabgabe....774
Versteckte Migrationshilfe für D81–D84....774
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....775
Datensicherung....776
Drucken und PDF....776
Sprache und Zahlenformat....777
Grenzen und Hinweise....777
Technische Einordnung in JUMBO BLOCK Rescue....777
Weiterführende Dokumentation....778
D.9 – Calc....779
Zweck....779
Rolle innerhalb der Engineering Suite....779
Wesentliche Funktionen....780
Grundrechner....780
Wissenschaftliche Funktionen....780
Speicherfunktionen....781
Prozentrechnung....781
Geometrie-Rechner....781
JUMBO BLOCK Flächenbelegung....782
Einheitenumrechnung....782
JUMBO BLOCK Modulrechner....783
Berechnungsverlauf....783
Daten und Datenaustausch....784
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....784
Datensicherung....784
PDF-Report....785
Sprache und Bedienung....785

Grenzen und Hinweise....785
Versionshinweis der Baseline....786
Technische Einordnung in JUMBO BLOCK Rescue....786
Weiterführende Dokumentation....786
D.10 – Units....788
Zweck....788
Rolle innerhalb der Engineering Suite....788
Wesentliche Funktionen....788
Kategorien....788
Mechanik & Physik....789
Elektrotechnik....789
Thermodynamik....789
Sonstiges....790
Umrechnung durchführen....790
Zahlenformatierung....790
Sonderfall Temperatur....791
Sonderfall Kraftstoffverbrauch....791
Berechnungsverlauf....791
Daten und Datenaustausch....792
Import....792
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....792
Datensicherung....793
PDF-Report....793
Sprache....794
Grenzen und Hinweise....794
Technische Einordnung in JUMBO BLOCK Rescue....794
Weiterführende Dokumentation....795
D.11 – Note....796
Zweck....796
Rolle innerhalb der Engineering Suite....796
Wesentliche Funktionen....796
Notizen erstellen und bearbeiten....796
Notizenliste....797
Suche....797
Speichern....797
Automatisches Speichern....798
Links und Dateien....798
Links....798
Dateien....798
Daten und Datenaustausch....799
Import....799
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....799
Datensicherung....800
Drucken und PDF....801
Sprache....801
Grenzen und Hinweise....801
Technische Einordnung in JUMBO BLOCK Rescue....802
Weiterführende Dokumentation....802
D.12 – Tables....803
Zweck....803
Rolle innerhalb der Engineering Suite....803

Arbeitsmappe und Arbeitsblätter....804
Formeln und Berechnungen....805
Formelfunktionen....805
Fehlererkennung....807
Bearbeitung und Zwischenablage....807
Rückgängig und Wiederholen....808
Zeilen und Spalten....808
Sortieren, Suchen und Ersetzen....809
Zellformatierung....809
Bedingte Formatierung....810
Fixieren von Zeilen und Spalten....810
Diagramme....810
Daten und Datenaustausch....811
Natives .jbtab-Format....811
XLSX....812
CSV....813
Datenintegrität....813
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....814
Datensicherung....814
Drucken und PDF....814
Sprache....815
Grenzen und Hinweise....816
Technische Einordnung in JUMBO BLOCK Rescue....816
Weiterführende Dokumentation....817
D.13 – Writer....818
Zweck....818
Rolle innerhalb der Engineering Suite....818
Dokumentaufbau....819
Briefkopf....819
Haupttext....819
Fußfeld....819
Seitenfuß....819
Textformatierung....820
Schriftarten....821
Serifenlos....821
Serifen....821
Monospace....821
Überschriften und Inhaltsverzeichnis....821
Automatische Felder....822
Tabellen....822
Bilder....823
Links und Dateien....823
Seiteneinstellungen....823
Vorlagen....824
Geschäftsbrief....824
Aktenvermerk....825
Protokoll....825
Suchen und Ersetzen....825
Wörter- und Zeichenzählung....826
Daten und Datenaustausch....826
Import und Sicherheitsbereinigung....826

Word-Export....827
Word-Import....828
ODT-Import....829
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....829
Datensicherung....830
PDF-Export....830
Datenintegrität....830
Ansicht und Bedienung....831
Sprache....831
Grenzen und Hinweise....832
Technische Einordnung in JUMBO BLOCK Rescue....832
Weiterführende Dokumentation....833
D.14 – Reader....834
Zweck....834
Rolle innerhalb der Engineering Suite....834
Unterstützte Dateiformate....835
EPUB....835
PDF....835
Markdown....835
Text....836
HTML/HTM....836
JSON und XML....836
Dokument öffnen....836
Dokumentbezogene Einstellungen....836
Navigation....837
EPUB....837
PDF....837
Textformate....837
Inhaltsverzeichnis und Seitenleiste....837
EPUB....837
PDF....837
Interne und externe Links....838
Darstellung....838
Lesebreite....838
Helle und dunkle Darstellung....838
Fokusmodus....839
Vollbildmodus....839
Suche....839
PDF....839
EPUB und Textformate....840
Grenze der PDF-Suche....840
Lesezeichen....840
PDF....840
EPUB....840
Textformate....840
Markierungen....840
Tastaturbedienung....841
Touch-Bedienung....842
Daten und Datenaustausch....842
Lokale Speicherung....842
Datenschutz....843

Bedeutung für JUMBO BLOCK Rescue....843
Grenzen und Hinweise....844
Technische Einordnung in JUMBO BLOCK Rescue....844
Weiterführende Dokumentation....845
D.15 – Text....846
Zweck....846
Rolle innerhalb der Engineering Suite....846
Wesentliche Funktionen....847
Dateien öffnen....847
Neue Datei....847
Bearbeiten....847
Syntaxhervorhebung....847
Zeilennummern....848
Suchen und Ersetzen....848
Rückgängig und Wiederholen....849
Daten und Dateiformate....849
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....849
Datensicherung....850
Sprache....850
Grenzen und Hinweise....850
Technische Einordnung in JUMBO BLOCK Rescue....851
Weiterführende Dokumentation....851
D.16 – PDF Merge....853
Zweck....853
Rolle innerhalb der Engineering Suite....853
PDFs hinzufügen....853
Dateiauswahl....854
Drag-and-drop....854
Laden der PDF-Dateien....854
Gemeinsame Seitenliste....854
Seiten anordnen....855
Eine Position verschieben....855
Freies Verschieben....855
Seiten entfernen....855
Speichern....856
Speicherort und Browserunterschiede....856
Chrome und Edge....856
Firefox und Safari....856
Datenintegrität....857
Zurücksetzen....857
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....857
Sitzungsbezogene Arbeitsweise....858
Sprache....858
Grenzen und Hinweise....859
Keine Seitenvorschau....859
Keine Bearbeitung von Seiteninhalten....859
Keine verschlüsselten PDF-Dateien....859
Keine dauerhafte Sitzung....859
Browserabhängiger Speicherort....859
Technische Einordnung in JUMBO BLOCK Rescue....859
Weiterführende Dokumentation....860

D.17 – PDF Printer....861
Zweck....861
Rolle innerhalb der Engineering Suite....861
PDF öffnen....862
Interaktive Vorschau....862
Zoom und Seitenanpassung....862
Vergrößern und Verkleinern....862
100 %....862
Breite....863
Seite....863
Stufenloser Zoom....863
Papierformat....863
Seitenausrichtung....863
Druckqualität....863
Entwurf....863
Gut....864
Hoch....864
Seiten pro Blatt – N-up....864
Broschürendruck....864
Duplex-Hinweis....865
Druckausgabe....865
Tastaturbedienung....865
Seitennavigation....865
Zoom....866
An Breite anpassen....866
100 %....866
Drucken....866
Daten und Datenaustausch....866
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....867
Bedeutung des Systemdruckers....867
Datenschutz....867
Technische Grundlage....868
Grenzen und Hinweise....868
Technische Einordnung in JUMBO BLOCK Rescue....869
Weiterführende Dokumentation....869
D.18 – Convert....870
Zweck....870
Rolle innerhalb der Engineering Suite....870
Bedienung....871
1. PDF auswählen....871
2. Konvertieren....871
3. Word-Datei speichern....871
Statusanzeige....871
Wie die Umwandlung funktioniert....872
Textextraktion....872
Kopf- und Fußzeilenerkennung....872
Absatzrekonstruktion....872
Tabellenerkennung....873
Bilder....873
Schriftart-Zuordnung....873
Daten und Datenaustausch....874

Speicherort wählen....874
Chrome und Edge....874
Firefox und Safari....875
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....875
Keine dauerhafte Arbeitssitzung....875
Sprache....875
Grenzen und Hinweise....876
Keine pixelgenaue 1:1-Konvertierung....876
Mehrspaltige Layouts....876
Kurze Tabellen....876
Kopf- und Fußzeilen....876
Vektorgrafiken....877
Keine OCR....877
Technische Einordnung in JUMBO BLOCK Rescue....877
Weiterführende Dokumentation....878
D.19 – Archive....879
Zweck....879
Rolle innerhalb der Engineering Suite....879
Unterstützte Formate....880
Technische Grundlage der Archivformate....880
Dateien und Ordner packen....880
Dateiliste....881
Archivname....881
Kompression....881
Dateinamen und Zeitangaben....882
ZIP64....882
Passwortschutz mit AES-256....883
Kompatibilität verschlüsselter ZIP-Dateien....883
Sichtbare Metadaten bei ZIP-Verschlüsselung....883
Umgang mit Passwörtern....884
Archive öffnen und prüfen....884
Alles entpacken....884
Chrome und Edge....884
Firefox und Safari....885
Zielordner und Browser-Schutz....885
Einzelne Dateien entpacken....885
Archive in ZIP umwandeln....886
Integritätsprüfung beim Entpacken....886
Sichere Pfadbehandlung....886
Passwortgeschützte Archive öffnen....887
Grenzen verschlüsselter 7z- und RAR-Archive....887
Fortschritt und Abbrechen....887
Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue....888
Bedeutung für gerettete Daten....888
Arbeitsspeicher und Archivgröße....889
ZIP, TAR und GZIP....889
7z, RAR, TAR.XZ und TAR.BZ2....889
Änderungsdaten beim Entpacken....889
Nicht unterstützte Archivvarianten....890
Seltene ZIP-Kompressionsverfahren....890
Mobile Nutzung....890

Sprache....890
Grenzen und Sicherheitshinweise....891
Technische Einordnung in JUMBO BLOCK Rescue....891
Weiterführende Dokumentation....892
D.20 – Files....893
Zweck....893
Rolle innerhalb der Engineering Suite....893
Browser und Zugriffsrechte....894
Chrome, Edge und Opera am Computer....894
Firefox, Safari und die meisten Smartphones....894
Lesen und Schreiben / Nur lesen....895
Geschützte Bereiche....895
Ordner öffnen....896
Zuletzt geöffnete Ordner....896
Navigation....897
Dateiliste....897
Listen- und Kachelansicht....897
Dateien und Ordner auswählen....898
Dateien und Ordner öffnen....898
Dateivorschau....898
Statusleiste und Ordnergröße....899
Neuen Ordner anlegen....899
Neue Textdatei anlegen....899
Umbenennen....899
Kopieren....900
Verschieben....900
Kopieren zwischen Datenträgern....901
Namenskonflikte....901
Schutz vor rekursivem Kopieren....902
Drag-and-drop von außen....902
Fortschritt und Abbrechen....902
Löschen....902
Suchen....903
Suche nach Namen....903
Suche nach Typ....903
Suche in Textinhalten....904
Suchergebnisse....904
Dateien herunterladen....904
Start vom USB-Stick....905
Lokale Arbeitsweise und Datenschutz....905
Bedeutung für JUMBO BLOCK Rescue....906
Dateizeitstempel und Attribute....906
Verschieben....907
Leistungsgrenzen....907
Gesperrte Dateien....907
Sprache und Darstellung....907
Grenzen und Sicherheitshinweise....908
Wichtige Tastenkombinationen....908
Technische Einordnung in JUMBO BLOCK Rescue....909
Weiterführende Dokumentation....910
Die Reise geht weiter....911

Block A

Block A – JB-Rescue bildet den funktionalen Kern von JUMBO BLOCK Rescue.

Die enthaltenen Werkzeuge dienen der Untersuchung, Diagnose, Sicherung und Wiederherstellung von Daten und Systemen. Sie decken unterschiedliche Rescue-Szenarien ab – von beschädigten Datenträgern und verlorenen Dateien über Netzwerk- und Sicherheitsanalysen bis zur Hardwarediagnose und technischen Bearbeitung geretteter Dateien.

Die Werkzeuge sind dabei nicht auf eine bestimmte Betriebssystemfamilie beschränkt. Ihre konkrete Einsetzbarkeit hängt vom jeweiligen Programm, Dateisystem, Datenträger, Protokoll oder Zielsystem ab.

Grundsätzlich gilt:

- **Zuerst identifizieren und diagnostizieren.**
- **Originale und beschädigte Quellen möglichst unverändert erhalten.**
- **Bei instabilen Datenträgern zuerst sichern.**
- **Analysen und Reparaturen bevorzugt an Images oder Arbeitskopien durchführen.**
- **Schreibende und destruktive Operationen nur bewusst und nach eindeutiger Identifikation von Quelle und Ziel ausführen.**

Werkzeuggruppen

35 Rescue-Werkzeuge in zehn Funktionsgruppen:

A.1–A.5 – Datenrettung
TestDisk, PhotoRec, GNU ddrescue, Foremost und Extundelete zur Sicherung beschädigter Datenträger sowie zur Wiederherstellung verlorener Partitionen und Dateien.

A.6–A.9 – Datenträger
GParted, GNOME Disks, SMART Diagnose und NVMe Diagnose zur Untersuchung, Verwaltung und Zustandsbewertung von Datenträgern und Partitionen.

A.10–A.14 – Sicherheit
ClamAV Scan, ClamAV Signaturen, Chkrootkit, Rkhunter und GUFW Firewall zur Malware-Prüfung, Systemanalyse und kontrollierten Netzwerkabsicherung.

A.15–A.16 – Analyse
Binwalk und Hexedit zur Untersuchung binärer Daten, Dateistrukturen, Firmware-Abbilder und unbekannter beziehungsweise beschädigter Dateien.

A.17–A.24 – Netzwerk
Nmap, Wireshark, TCPDump, ARP Scan, MTR, Dig, Whois und Chromium zur Netzwerkdiagnose, Paketanalyse, Erreichbarkeitsprüfung, Namensauflösung und kontrollierten Recherche.

A.25–A.26 – Verschlüsselung
Cryptsetup und GnuPG für den kontrollierten Zugriff auf verschlüsselte Datenträger sowie für kryptografische Datei-, Schlüssel- und Signaturaufgaben.

A.27–A.28 – Remote
SSH und SSHFS für verschlüsselte Remote-Verbindungen sowie den Zugriff auf entfernte Dateisysteme während Rescue- und Sicherungsarbeiten.

A.29–A.33 – Hardware
HWInfo, Inxi, LSHW, DMI Decode und Sensoren zur Identifikation und Diagnose der vorhandenen Hardware und ihrer Betriebsparameter.

A.34 – Dateien
Midnight Commander zur lokalen und entfernten Dateiverwaltung sowie zum kontrollierten Kopieren, Verschieben und Organisieren von Rescue-Daten.

A.35 – Dokumente
QPDF zur technischen Prüfung, strukturellen Bearbeitung und gegebenenfalls Wiederherstellung beschädigter PDF-Dateien.

Die 35 Werkzeuge ergänzen sich innerhalb eines Rescue-Falls. Nicht jedes Programm ist für jedes Zielsystem geeignet und nicht jede erkannte Beschädigung sollte unmittelbar repariert werden.

Der grundlegende Arbeitsablauf von JUMBO BLOCK Rescue lautet deshalb:

Identifizieren → Zustand beurteilen → Original schützen → erforderlichenfalls sichern oder ein Image erstellen → Arbeitskopie untersuchen → gezielt retten oder reparieren → Ergebnis kontrollieren.

JUMBO BLOCK Rescue ist kein Werkzeug für ein einzelnes Betriebssystem, sondern eine unabhängige Rescue-Umgebung für die Untersuchung und Rettung unterschiedlichster Systeme, Datenträger, Dateien und Netzwerkumgebungen.

Zuordnung der Bereiche, Programme und Versionen

Nr.	Bereich	Programm	Baseline-Version
A.1	Datenrettung	TestDisk	7.2 ff.
A.2	Datenrettung	PhotoRec	7.2 ff.
A.3	Datenrettung	GNU ddrescue	1.29 ff.
A.4	Datenrettung	Foremost	1.5.7 ff.
A.5	Datenrettung	Extundelete	0.2.4 ff.
A.6	Datenträger	GParted	1.6.0 ff.
A.7	Datenträger	Laufwerke / GNOME Disks	46.1 ff.
A.8	Datenträger	SMART Diagnose	smartmontools 7.4-3
A.9	Datenträger	NVMe Diagnose	nvme-cli 2.13 ff.
A.10	Sicherheit	ClamAV Scan	1.4.3 ff.
A.11	Sicherheit	ClamAV Signaturen	1.4.3 ff.
A.12	Sicherheit	Chkrootkit	0.58b ff.
A.13	Sicherheit	Rkhunter	rkhunter 1.4.6-13
A.14	Sicherheit	GUFW Firewall	24.04.0 ff.
A.15	Analyse	Binwalk	2.4.3 ff.
A.16	Analyse	Hexedit	1.6 ff.
A.17	Netzwerk	Nmap	7.95 ff.
A.18	Netzwerk	Wireshark	4.4.18
A.19	Netzwerk	TCPDump	4.99.5 ff.
A.20	Netzwerk	ARP Scan	1.10.0 ff.
A.21	Netzwerk	MTR	0.95 ff.
A.22	Netzwerk	Dig	bind9-dnsutils 1:9.20.29-1
A.23	Netzwerk	Whois	5.6.3
A.24	Netzwerk	Chromium	154.0.8037.57 ff.
A.25	Verschlüsselung	Cryptsetup	2.7.5 ff.
A.26	Verschlüsselung	GnuPG	2.4.7 ff.
A.27	Remote	SSH	openSSH 10.0p1 ff.

Nr.	Bereich	Programm	Baseline-Version
A.28	Remote	SSHFS	sshfs 3.7.3-1.2
A.29	Hardware	HWInfo	21.82 ff.
A.30	Hardware	Inxi	3.3.38 ff.
A.31	Hardware	LSHW	lshw 02.19.git.2021.06... ff.
A.32	Hardware	DMI Decode	dmidecode 3.6 ff.
A.33	Hardware	Sensoren	lm-sensors 3.6.2 ff.
A.34	Dateien	Midnight Commander	4.8.33 ff.
A.35	Dokumente	QPDF	12.2.0-1 ff.

A.1 TestDisk

Bereich: Datenrettung
Programm: TestDisk
Baseline 20261001: TestDisk 7.2
Installiertes Paket: `testdisk 7.2-0.1`
Bedienung: textbasierte, interaktive Benutzeroberfläche im Terminal

Zweck

TestDisk ist ein freies Open-Source-Werkzeug zur Wiederherstellung verlorener Partitionen, zur Reparatur bestimmter Dateisystemstrukturen und zur Wiederherstellung gelöschter Dateien. In JUMBO BLOCK Rescue gehört TestDisk deshalb zu den zentralen Werkzeugen für die logische Datenrettung.

Die Baseline 20261001 enthält das Paket `testdisk` in Version `7.2-0.1`. Die offizielle Dokumentation für TestDisk 7.2 nennt als wesentliche Einsatzgebiete die Wiederherstellung verlorener Partitionen, die Reparatur von Bootsektoren sowie die Wiederherstellung von Dateien aus beschädigten oder gelöschten Dateisystemstrukturen.

TestDisk arbeitet dabei nicht wie ein gewöhnlicher Dateimanager. Es analysiert die Strukturen eines Datenträgers und kann Informationen rekonstruieren, die durch eine beschädigte oder gelöschte Partitionstabelle nicht mehr regulär erreichbar sind.

Typische Einsatzfälle

TestDisk kann in JUMBO BLOCK Rescue insbesondere eingesetzt werden, wenn eine Partition versehentlich gelöscht wurde, eine Partitionstabelle beschädigt ist, ein vorhandenes Dateisystem vom Betriebssystem nicht mehr regulär erreicht werden kann oder bestimmte Bootsektorstrukturen beschädigt wurden.

Darüber hinaus besitzt TestDisk Funktionen zur Wiederherstellung gelöschter Dateien. Die offizielle Dokumentation nennt hierfür FAT12/16/32, exFAT, NTFS und ext2. Bei ext3 und ext4 kann TestDisk zwar Dateisystemstrukturen analysieren, die direkte Wiederherstellung gelöschter Dateiinhalte über die Undelete-Funktion ist jedoch aufgrund der dort verfügbaren Metadaten nicht entsprechend möglich.

Start und grundlegende Bedienung

TestDisk wird innerhalb von JUMBO BLOCK Rescue im Terminal ausgeführt. Die Benutzeroberfläche ist vollständig tastaturgesteuert.

Nach dem Start bietet TestDisk zunächst die Möglichkeit, eine Protokolldatei anzulegen. Die offizielle Dokumentation empfiehlt normalerweise **Create**. Dabei wird im aktuellen Verzeichnis eine Datei mit dem Namen `testdisk.log` angelegt.

Anschließend wird der zu untersuchende Datenträger ausgewählt. TestDisk zeigt erkannte Laufwerke mit ihrer Größe an. Danach folgt die Auswahl beziehungsweise Erkennung des verwendeten Partitionstabellentyps. TestDisk versucht den passenden Typ automatisch zu

erkennen; nach der offiziellen Anleitung ist der vorgeschlagene Standardwert üblicherweise korrekt.

Analyse einer Partitionstabelle

Für die Suche nach verlorenen Partitionen wird der Menüpunkt **Analyse** verwendet.

TestDisk zeigt zunächst die aktuell erkannte Partitionstabelle an. Danach kann mit **Quick Search** nach vorhandenen beziehungsweise verlorenen Partitionen gesucht werden. Gefundene Partitionen werden bereits während des Suchvorgangs angezeigt.

Eine wichtige Kontrollmöglichkeit besteht darin, eine gefundene Partition zu markieren und mit der Taste **P** deren Dateien anzeigen zu lassen. Diese Prüfung hilft festzustellen, ob tatsächlich die gesuchte Partition gefunden wurde. Die Dateiansicht wird für FAT, exFAT, NTFS sowie ext2/3/4 unterstützt.

Wird die gesuchte Partition mit **Quick Search** nicht gefunden, steht **Deeper Search** zur Verfügung. Diese tiefere Suche untersucht den Datenträger umfassender und kann entsprechend erheblich länger dauern.

Wiederherstellung einer Partition

Gefundene Partitionen werden von TestDisk mit ihrem jeweiligen Status dargestellt. Als gelöscht gekennzeichnete Partitionen werden nicht automatisch wiederhergestellt.

Der Anwender muss daher prüfen, welche gefundenen Partitionen tatsächlich zur ursprünglichen Datenträgerstruktur gehören. Erst nachdem Größe, Lage und möglichst auch die enthaltenen Dateien überprüft wurden, kann der Status einer wiederherzustellenden Partition entsprechend gesetzt werden.

Mit **Write** kann die ausgewählte Partitionsstruktur anschließend in die Partitionstabelle geschrieben werden. Dieser Schritt verändert den untersuchten Datenträger. Die offizielle Dokumentation weist ausdrücklich darauf hin, **Write nur dann zu bestätigen, wenn die angezeigte Partitionsstruktur vollständig geprüft wurde und alle gewünschten Partitionen korrekt aufgeführt sind**.

Wiederherstellung gelöschter Dateien

Neben vollständigen Partitionen kann TestDisk unter unterstützten Dateisystemen auch einzelne gelöschte Dateien beziehungsweise Verzeichnisse auffinden.

Der entsprechende Arbeitsweg führt über **Advanced** und anschließend über **Undelete** für die betreffende Partition. Gefundene gelöschte Dateien werden angezeigt und können ausgewählt und auf ein Zielmedium kopiert werden.

Entscheidend ist dabei die Wahl des Zielmediums: Wiederhergestellte Dateien sollten **nicht auf dasselbe Dateisystem zurückgeschrieben werden, von dem sie gerettet werden**. Jeder Schreibvorgang auf dem betroffenen Datenträger kann bislang noch vorhandene gelöschte Daten überschreiben und damit eine weitere Wiederherstellung verhindern. Die

TestDisk-Dokumentation empfiehlt deshalb ausdrücklich ein anderes Zielmedium beziehungsweise mindestens ein anderes Dateisystem.

Bootsektoren und Dateisystemstrukturen

TestDisk enthält zusätzlich Funktionen zur Reparatur bestimmter Dateisystemstrukturen. Die Dokumentation von Version 7.2 beschreibt unter anderem die Reparatur beziehungsweise Rekonstruktion von FAT-, exFAT- und NTFS-Bootsektoren sowie die Arbeit mit ext2/3/4-Superblocks.

Bei FAT32 und NTFS kann beispielsweise ein vorhandener Backup-Bootsektor zur Wiederherstellung des Haupt-Bootsektors verwendet werden. TestDisk kann hierzu den Backup-Bootsektor erkennen und über **Backup BS** auf den Haupt-Bootsektor übertragen.

Diese Funktionen greifen direkt in Dateisystemstrukturen ein und sollten daher erst nach einer vorherigen Analyse und Sicherung eingesetzt werden.

Vorgehen bei physisch beschädigten Datenträgern

TestDisk ist primär ein Werkzeug für die **logische** Datenrettung. Bei Datenträgern mit Lesefehlern oder defekten Sektoren sollte nicht unnötig lange direkt auf dem beschädigten Original gearbeitet werden.

Die offizielle TestDisk-Dokumentation empfiehlt bei beschädigten Datenträgern, zunächst eine Kopie beziehungsweise ein Image zu erstellen und anschließend auf diesem Klon weiterzuarbeiten.

JUMBO BLOCK Rescue enthält dafür mit **GNU ddrescue** ein separates Werkzeug, das in Abschnitt **A.3** ausführlich behandelt wird.

Damit ergibt sich für einen physisch auffälligen Datenträger grundsätzlich die sicherere Arbeitsreihenfolge:

Originaldatenträger → sektorweise Sicherung/Abbild → Analyse des Abbilds → Wiederherstellung der Daten

TestDisk kann auch direkt mit Datenträgerabbildern arbeiten. Die offizielle Dokumentation führt die Arbeit mit Disk Images ausdrücklich als unterstützte Betriebsart auf.

TestDisk und PhotoRec

TestDisk und PhotoRec gehören zum selben Datenrettungspaket, verfolgen aber unterschiedliche Ansätze.

TestDisk arbeitet primär mit vorhandenen Partitionierungs- und Dateisystemstrukturen. Dadurch können bei erfolgreicher Wiederherstellung beispielsweise ursprüngliche Partitionen, Verzeichnisstrukturen und Dateinamen erhalten bleiben.

PhotoRec arbeitet dagegen signaturbasiert und sucht unabhängig von einer intakten Dateisystemstruktur nach bekannten Dateitypen. Wenn TestDisk verlorene Dateien nicht

mehr über die vorhandenen Dateisysteminformationen rekonstruieren kann, empfiehlt auch die offizielle Dokumentation den Einsatz von PhotoRec.

PhotoRec wird im folgenden Abschnitt **A.2** separat behandelt.

Sicherheitshinweise

Bei einer Datenrettung ist der Schutz des noch vorhandenen Datenbestands wichtiger als eine möglichst schnelle Reparatur.

Auf einen betroffenen Datenträger sollten deshalb möglichst keine neuen Daten geschrieben werden. Insbesondere das Formatieren, Neuinstallieren eines Betriebssystems oder Kopieren neuer Dateien auf den Datenträger kann noch wiederherstellbare Daten überschreiben.

Besondere Vorsicht ist bei TestDisk-Funktionen erforderlich, die tatsächlich Änderungen auf dem Datenträger durchführen. Dazu gehört insbesondere das Schreiben einer neuen Partitionstabelle über **Write** sowie das Überschreiben beziehungsweise Rekonstruieren von Bootsektoren.

Solange lediglich analysiert und gesucht wird, kann zunächst geprüft werden, welche Strukturen TestDisk erkennt. Schreibende Reparaturschritte sollten erst nach eindeutiger Identifikation des Datenträgers und der wiederherzustellenden Strukturen erfolgen.

Rolle in JUMBO BLOCK Rescue

TestDisk bildet innerhalb von JUMBO BLOCK Rescue die erste Werkzeugstufe für die **strukturorientierte Datenrettung**.

Es ist besonders geeignet, wenn die eigentlichen Daten vermutlich noch vorhanden sind, der Zugriff jedoch durch eine verlorene Partition, eine beschädigte Partitionstabelle oder bestimmte beschädigte Dateisystemstrukturen verhindert wird.

Bei einem technisch instabilen Datenträger sollte dagegen zunächst GNU ddrescue zur Sicherung des Originals eingesetzt werden. Ist die Dateisystemstruktur so stark beschädigt, dass eine strukturorientierte Wiederherstellung nicht mehr möglich ist, steht mit PhotoRec zusätzlich eine signaturbasierte Datenrettung zur Verfügung.

Damit ergänzen sich die ersten drei Datenrettungswerkzeuge von JUMBO BLOCK Rescue funktional:

TestDisk – Strukturen analysieren und rekonstruieren
PhotoRec – Dateien anhand ihrer Inhalte wiederfinden
GNU ddrescue – beschädigte Datenträger möglichst schonend sichern

A.2 PhotoRec

Bereich: Datenrettung
Programm: PhotoRec
Baseline 20261001: PhotoRec 7.2
Installiertes Paket: `testdisk 7.2-0.1`
Bedienung: textbasierte, interaktive Benutzeroberfläche im Terminal

Zweck

PhotoRec ist ein Datenrettungsprogramm zur signaturbasierten Wiederherstellung von Dateien. Es gehört zum TestDisk-Paket, verfolgt bei der Datenrettung jedoch einen anderen Ansatz als TestDisk.

In der JUMBO BLOCK Rescue Baseline 20261001 ist das Paket `testdisk` in Version `7.2-0.1` installiert. Dieses Paket stellt neben TestDisk auch PhotoRec bereit.

Der Hersteller CGSecurity bezeichnet PhotoRec als **File Carver**. Das Programm sucht nach bekannten Dateisignaturen und kann dadurch Dateien auch dann wiederfinden, wenn das zugrunde liegende Dateisystem beschädigt ist. PhotoRec rekonstruiert dabei grundsätzlich nicht die ursprüngliche Verzeichnisstruktur und nicht die ursprünglichen Dateinamen.

Typische Einsatzfälle

PhotoRec kommt insbesondere dann zum Einsatz, wenn eine klassische Wiederherstellung über vorhandene Dateisystemstrukturen nicht mehr möglich oder nicht erfolgreich ist.

Typische Anwendungsfälle sind gelöschte Dateien, beschädigte Dateisysteme, formatierte Partitionen sowie Datenträger, bei denen die ursprüngliche Dateisystemstruktur nicht mehr ausreichend verwertbar ist.

CGSecurity nennt unter den von PhotoRec erkannten Daten unter anderem ZIP-Archive, Office-Dokumente, PDF- und HTML-Dateien, JPEG-Bilder und zahlreiche weitere Grafik- und Dateiformate. PhotoRec 7.2 unterstützt laut Hersteller mehr als **300 Dateifamilien mit mehr als 480 Dateierweiterungen**.

Arbeitsprinzip

PhotoRec verlässt sich bei der eigentlichen Dateirettung nicht auf die ursprünglichen Verzeichniseinträge und Dateinamen.

Stattdessen untersucht es die Datenbereiche nach charakteristischen Signaturen bekannter Dateiformate. Wird eine solche Signatur erkannt, versucht PhotoRec, die zugehörigen Daten als Datei zu rekonstruieren.

Dieses Verfahren hat einen wesentlichen Vorteil: Auch ein stark beschädigtes Dateisystem kann noch verwertbare Dateien enthalten.

Der Nachteil besteht darin, dass die ursprüngliche logische Organisation normalerweise verloren geht. Wiederhergestellte Dateien erhalten daher nicht automatisch ihre ursprünglichen Dateinamen und Verzeichnispfade zurück.

Start und Auswahl des Datenträgers

PhotoRec wird unter Linux im Terminal ausgeführt. Für den direkten Zugriff auf physische Datenträger sind entsprechende Zugriffsrechte erforderlich; die Dokumentation für PhotoRec 7.2 beschreibt hierfür unter Linux die Ausführung mit Root-Rechten.

Nach dem Start zeigt PhotoRec die verfügbaren Datenträger an.

Der zu untersuchende Datenträger wird mit den Pfeiltasten ausgewählt und anschließend bestätigt. Danach wird die Partition ausgewählt, in der nach verlorenen Dateien gesucht werden soll.

Vor Beginn einer Wiederherstellung muss sorgfältig geprüft werden, dass tatsächlich der richtige Quell-Datenträger ausgewählt wurde.

Auswahl der Quelldaten

Nach Auswahl einer Partition stellt PhotoRec unter anderem folgende Funktionen bereit:

Search startet die Datenrettung.

Options öffnet zusätzliche Einstellungen für das Wiederherstellungsverfahren.

File Opt bestimmt, nach welchen Dateitypen PhotoRec suchen soll.

Gerade **File Opt** ist bei großen Datenträgern nützlich, wenn gezielt bestimmte Dateiformate gesucht werden sollen.

Der Hersteller weist allerdings darauf hin, dass das Abschalten zu vieler Dateiformate unerwartete Auswirkungen haben kann. Bei Dateiformaten, deren Größe nicht unmittelbar aus dem Dateikopf bestimmt werden kann, verwendet PhotoRec unter anderem das Auftreten einer neuen bekannten Dateisignatur zur Abgrenzung. Werden zu viele Signaturen deaktiviert, können dadurch übergroße rekonstruierte Dateien entstehen.

Auswahl des Dateisystemtyps

Vor der eigentlichen Suche fragt PhotoRec nach dem Dateisystemtyp.

Für **ext2/ext3/ext4** wird die entsprechende ext-Dateisystemoption verwendet. Bei anderen unterstützten Dateisystemen wird gemäß Herstellerdokumentation **Other** gewählt.

Diese Auswahl dient PhotoRec dazu, die Organisation der Datenblöcke richtig zu berücksichtigen.

Gesamte Partition oder nur freier Speicher

PhotoRec kann je nach Dateisystem entweder die gesamte Partition oder ausschließlich den nicht belegten Speicherbereich untersuchen.

Die Suche über die **gesamte Partition** ist insbesondere bei einem beschädigten Dateisystem sinnvoll.

Die Suche ausschließlich im **freien beziehungsweise nicht zugeordneten Speicher** konzentriert sich dagegen auf gelöschte Dateien. Diese Möglichkeit steht laut PhotoRec-7.2-Dokumentation für ext2/ext3/ext4, FAT12/16/32 und NTFS zur Verfügung.

Die Auswahl beeinflusst damit unmittelbar Umfang und Ziel der Datenrettung.

Auswahl der wiederherzustellenden Dateitypen

Über **File Opt** können einzelne Dateitypen beziehungsweise Dateifamilien aktiviert oder deaktiviert werden.

Damit lässt sich beispielsweise eine Rettung auf bestimmte Dokument-, Bild-, Archiv- oder Multimediaformate konzentrieren.

PhotoRec besitzt darüber hinaus die Möglichkeit, eigene Dateisignaturen zu definieren. Hierzu kann unter Linux eine Datei `.photorec.sig` im Home-Verzeichnis beziehungsweise eine `photorec.sig` im aktuellen Verzeichnis verwendet werden. Dadurch können auch Dateiformate berücksichtigt werden, für die PhotoRec standardmäßig keine passende Signatur kennt.

Auswahl des Zielverzeichnisses

Vor Beginn der Wiederherstellung muss ein Zielverzeichnis für die geretteten Dateien ausgewählt werden.

Hier gilt eine der wichtigsten Regeln der gesamten Datenrettung:

Gerettete Dateien dürfen nicht auf das untersuchte Quelldateisystem geschrieben werden.

Neue Schreibvorgänge könnten Bereiche überschreiben, in denen sich noch nicht wiederhergestellte Daten befinden. Dadurch könnten bislang rettbare Dateien endgültig verloren gehen.

CGSecurity warnt ausdrücklich davor, das Quell-Dateisystem gleichzeitig als Ziel der Wiederherstellung zu verwenden.

Das Ziel sollte deshalb auf einem anderen Datenträger beziehungsweise einem geeigneten anderen Dateisystem liegen.

Der Hersteller weist außerdem darauf hin, dass FAT32 als Ziel problematisch sein kann, wenn Dateien mit mehr als 4 GB wiederhergestellt werden sollen.

Durchführung der Wiederherstellung

Nach dem Start untersucht PhotoRec die ausgewählten Datenbereiche.

Während des Vorgangs wird die Anzahl der bereits wiederhergestellten Dateien laufend angezeigt. Die geretteten Dateien werden in Verzeichnissen mit Namen wie

```
recup_dir.1
recup_dir.2
recup_dir.3
```

gespeichert.

Auf diese Dateien kann bereits während einer noch laufenden Wiederherstellung zugegriffen werden.

PhotoRec führt mehrere Verarbeitungsschritte beziehungsweise Durchläufe durch. In der ersten Phase kann unter anderem die Blockgröße bestimmt werden; anschließend erfolgt die eigentliche Wiederherstellung erkannter Dateien.

Dateinamen und Verzeichnisstruktur

Da PhotoRec die Dateien anhand ihrer Inhalte und Signaturen rekonstruiert, stehen ursprüngliche Dateinamen und Verzeichnisstrukturen normalerweise nicht mehr zur Verfügung.

Die erzeugten Dateinamen beginnen standardmäßig mit einem Buchstaben und einer Nummer.

Dabei dokumentiert CGSecurity unter anderem:

`f` – wiederhergestellte Datei
`b` – beschädigte beziehungsweise als beschädigt erhaltene Datei
`t` – eingebettetes JPEG-Vorschaubild

PhotoRec kann bei bestimmten Dateiformaten vorhandene Metadaten verwenden. Dadurch können beispielsweise Informationen aus Dokumenten oder Zeitinformationen aus Bildern zur Benennung beziehungsweise zeitlichen Zuordnung herangezogen werden. Der Hersteller warnt jedoch ausdrücklich davor, solche Zeitangaben bei forensischen Untersuchungen ungeprüft als zuverlässig anzusehen.

Optionen für schwierige Fälle

PhotoRec 7.2 besitzt mehrere vom Hersteller dokumentierte Optionen für besondere Rettungssituationen.

Paranoid überprüft wiederhergestellte Dateien und verwirft standardmäßig als ungültig erkannte Ergebnisse.

Eine zusätzliche **Bruteforce**-Option kann zur Wiederherstellung stärker fragmentierter JPEG-Dateien eingesetzt werden. Diese Methode ist laut Hersteller sehr rechenintensiv.

Keep corrupted files ermöglicht es, auch als beschädigt erkannte Dateien beziehungsweise Dateifragmente zu behalten. Diese können anschließend möglicherweise mit anderen Werkzeugen weiter untersucht werden.

Low memory reduziert den Speicherbedarf für Systeme, bei denen PhotoRec während einer Wiederherstellung aufgrund unzureichenden Arbeitsspeichers Probleme bekommt. CGSecurity empfiehlt, diese Option nur einzusetzen, wenn sie tatsächlich erforderlich ist.

Der **Expert Mode** ermöglicht unter anderem die manuelle Vorgabe von Dateisystem-Blockgröße und Offset und richtet sich damit an spezielle Wiederherstellungssituationen.

Arbeiten mit Datenträgerabbildern

PhotoRec kann nicht nur physische Datenträger, sondern auch Datenträgerabbilder untersuchen.

Die Dokumentation für Version 7.2 nennt ausdrücklich **RAW-Abbilder (`.dd`)** sowie **EnCase-EWF-Abbilder (`.E01`)** und aufgeteilte EnCase-Abbilder. Auf Datenträgerabbildern benötigt PhotoRec keinen direkten administrativen Zugriff auf ein physisches Laufwerk.

Damit eignet sich PhotoRec besonders gut für die Arbeit mit zuvor erzeugten Sicherungsabbildern.

Bei einem physisch beschädigten oder instabilen Originaldatenträger sollte daher möglichst zunächst ein Abbild beziehungsweise Klon erstellt werden. Innerhalb von JUMBO BLOCK Rescue steht hierfür **GNU ddrescue** zur Verfügung.

Fragmentierte Dateien

File Carving besitzt technische Grenzen.

Insbesondere stark fragmentierte Dateien können problematisch sein, weil ihre Daten nicht zwangsläufig zusammenhängend auf dem Datenträger gespeichert sind.

CGSecurity beschreibt dies ausdrücklich am Beispiel bestimmter Videoaufzeichnungen von Digitalkameras. PhotoRec erwartet grundsätzlich nicht fragmentierte Dateien; bestimmte fragmentierte Videoformate können deshalb nach der Wiederherstellung zunächst unbrauchbar sein und zusätzliche Rekonstruktionsschritte erfordern.

Eine gefundene Dateisignatur garantiert daher nicht automatisch, dass eine vollständig intakte Datei wiederhergestellt werden kann.

Fortsetzen einer unterbrochenen Wiederherstellung

Wird eine laufende PhotoRec-Wiederherstellung unterbrochen, kann PhotoRec beim nächsten Start anbieten, den Vorgang fortzusetzen.

Nach Abschluss zeigt das Programm eine Zusammenfassung der Wiederherstellung an.

Sicherheitshinweise

Auf einem Datenträger mit verlorenen Daten sollten möglichst keine weiteren Schreibvorgänge stattfinden.

Jeder neue Schreibvorgang kann Datenblöcke überschreiben, die noch Bestandteile gelöschter Dateien enthalten.

Deshalb gilt für JUMBO BLOCK Rescue:

Quelle möglichst unverändert lassen.

Wiederhergestellte Dateien auf einen anderen Datenträger schreiben.

Bei einem physisch auffälligen Datenträger zunächst ein Sicherungsabbild erstellen.

Die eigentliche Datenrettung möglichst auf dem Abbild beziehungsweise Klon durchführen.

Diese Vorgehensweise reduziert zusätzliche Veränderungen am Originalmedium und entspricht den grundlegenden Sicherheitshinweisen der PhotoRec-Dokumentation.

TestDisk oder PhotoRec?

Ob TestDisk oder PhotoRec eingesetzt wird, hängt vom Zustand der Datenstrukturen ab.

TestDisk versucht vorhandene Partitionierungs- und Dateisystemstrukturen zu nutzen beziehungsweise bestimmte beschädigte Strukturen wiederherzustellen. Dadurch besteht die Möglichkeit, ursprüngliche Dateinamen und Verzeichnisstrukturen zu erhalten.

PhotoRec sucht dagegen anhand der Dateiinhalte beziehungsweise Dateisignaturen. Es kann deshalb auch bei beschädigten Dateisystemen noch Dateien finden, verliert dabei jedoch grundsätzlich die ursprüngliche Dateisystemorganisation.

CGSecurity empfiehlt bei verlorenen Partitionen beziehungsweise bestimmten gelöschten Dateien zunächst TestDisk, da eine erfolgreiche strukturorientierte Wiederherstellung schneller sein und ursprüngliche Dateinamen erhalten kann. PhotoRec ist insbesondere dann wichtig, wenn dieser Weg nicht mehr erfolgreich ist.

Rolle in JUMBO BLOCK Rescue

PhotoRec bildet innerhalb von JUMBO BLOCK Rescue die **signaturbasierte Datenrettung**.

Das Werkzeug ergänzt TestDisk insbesondere in Situationen, in denen Partitionierungs- oder Dateisystemstrukturen nicht mehr ausreichend für eine strukturorientierte Wiederherstellung genutzt werden können.

Gemeinsam mit GNU ddrescue ergibt sich eine abgestufte Rettungsstrategie:

TestDisk – Partitionierungs- und Dateisystemstrukturen untersuchen und, soweit unterstützt, wiederherstellen

PhotoRec – Dateien unabhängig von einer intakten Dateisystemstruktur anhand ihrer Signaturen rekonstruieren

GNU ddrescue – problematische Datenträger zunächst möglichst schonend auf ein anderes Medium oder in ein Abbild kopieren

Bei einem intakten Datenträger kann unmittelbar analysiert werden. Bei Anzeichen eines physischen Defekts hat dagegen die Sicherung des Originals Vorrang vor weiteren intensiven Analyse- oder Rettungsversuchen.

A.3 GNU ddrescue

Bereich: Datenrettung
Programm: GNU ddrescue
Baseline 20261001: GNU ddrescue 1.29
Installiertes Paket: `gddrescue 1.29-1`
Bedienung: Kommandozeile im Terminal

Zweck

GNU ddrescue ist ein Datenrettungswerkzeug zum Kopieren von Daten von einem Datei- oder Blockgerät auf ein anderes Ziel. Sein Schwerpunkt liegt darauf, bei Lesefehlern möglichst viele noch lesbare Daten zu retten und dabei Bereiche mit Problemen zunächst so zu behandeln, dass gut lesbare Bereiche nicht unnötig lange blockiert werden.

In JUMBO BLOCK Rescue übernimmt GNU ddrescue damit eine andere Aufgabe als TestDisk und PhotoRec: Es versucht zunächst, den **lesbaren Inhalt eines problematischen Datenträgers möglichst vollständig auf ein anderes Medium oder in eine Image-Datei zu übertragen**.

Die Baseline 20261001 enthält hierfür das Paket `gddrescue` in Version `1.29-1`.

GNU selbst beschreibt ddrescue als Datenrettungswerkzeug, das Daten von einer Datei oder einem Blockgerät auf ein anderes kopiert und bei Lesefehlern versucht, zunächst die guten Bereiche zu retten.

Abgrenzung zu dd und dd_rescue

GNU ddrescue darf nicht mit dem Unix-/GNU-Werkzeug `dd` verwechselt werden.

`dd` ist ein allgemeines Werkzeug zum Kopieren und Konvertieren von Daten. GNU Coreutils weist selbst darauf hin, dass für ausfallende Speichermedien Werkzeuge wie GNU ddrescue zusätzliche Funktionen besitzen, die darauf ausgelegt sind, vor dem endgültigen Ausfall eines Gerätes möglichst viele Daten zu sichern.

Ebenfalls nicht identisch ist GNU ddrescue mit dem separat entwickelten Programm `dd_rescue`.

Für dieses Handbuch bezeichnet **ddrescue ausschließlich GNU ddrescue 1.29** aus dem Baseline-Paket `gddrescue`.

Typische Einsatzfälle

GNU ddrescue ist insbesondere für Datenträger vorgesehen, bei denen beim Lesen Fehler auftreten.

Dazu können beispielsweise Festplatten, SSDs, USB-Datenträger, Speicherkarten, optische Medien oder einzelne Partitionen gehören, sofern das Betriebssystem sie noch als lesbares Blockgerät bereitstellt.

Ein typischer Rescue-Fall ist ein Datenträger, bei dem einzelne Bereiche noch problemlos gelesen werden können, andere Bereiche jedoch I/O-Fehler verursachen.

Statt sofort intensiv auf den problematischen Stellen zu arbeiten, versucht ddrescue zunächst, möglichst viel gut lesbares Material zu sichern.

Grundprinzip

Das zentrale Prinzip von GNU ddrescue besteht darin, bei einem beschädigten Datenträger nicht unnötig viel Zeit mit den schwierigsten Bereichen zu verbringen, solange andere Daten noch problemlos gelesen werden können.

Dadurch soll möglichst früh ein großer Anteil der noch verfügbaren Daten auf dem Zielmedium gesichert werden.

Problematische Bereiche können anschließend in weiteren Phasen genauer bearbeitet werden.

Das unterscheidet ddrescue wesentlich von einem einfachen linearen Kopiervorgang.

Quelle und Ziel

Ein ddrescue-Aufruf besteht grundsätzlich aus:

```
ddrescue [Optionen] Eingabedatei Ausgabedatei [Mapfile]
```

Als Eingabe kann beispielsweise ein physischer Datenträger oder eine Partition dienen.

Als Ausgabe kann entweder ein anderer Datenträger oder eine Image-Datei verwendet werden.

Beispiel für ein Image:

```
sudo ddrescue /dev/sdX rescue.img rescue.map
```

Dabei steht:

`/dev/sdX` für das Quellgerät,

`rescue.img` für die Zieldatei,

`rescue.map` für das Mapfile.

Die Gerätenamen in diesem Beispiel sind Platzhalter. Sie müssen vor einem realen Einsatz anhand des tatsächlich angeschlossenen Systems überprüft werden.

Eine Verwechslung von Quelle und Ziel kann zu schwerem Datenverlust führen.

Das Mapfile

Eine der wichtigsten Funktionen von GNU ddrescue ist das **Mapfile**.

Das Mapfile dokumentiert den Zustand des Rettungsvorgangs. Es hält fest, welche Bereiche bereits erfolgreich kopiert wurden und welche Bereiche noch nicht beziehungsweise nicht erfolgreich bearbeitet werden konnten.

Dadurch kann ein unterbrochener Rettungsvorgang später fortgesetzt werden, ohne bereits erfolgreich kopierte Bereiche erneut vollständig lesen zu müssen.

Für ernsthafte Rettungsvorgänge sollte deshalb ein Mapfile verwendet und aufbewahrt werden.

Das Mapfile ist nicht das eigentliche Datenträgerabbild. Es enthält vielmehr die Informationen, die ddrescue benötigt, um den Stand der Rettung zu verwalten.

Fortsetzen eines Rettungsvorgangs

Wird ddrescue beendet oder ein Rettungsvorgang unterbrochen, kann bei einem erneuten Aufruf dasselbe Mapfile verwendet werden.

GNU ddrescue kann anhand dieser Informationen erkennen, welche Bereiche bereits bearbeitet wurden.

Damit können Rettungsversuche über mehrere Sitzungen verteilt werden.

Das ist gerade bei instabilen oder langsam ausfallenden Datenträgern wichtig, da eine vollständige Rettung unter Umständen nicht in einem einzigen Durchlauf möglich ist.

Rettungsphasen

GNU ddrescue bearbeitet einen Datenträger nicht lediglich mit einem einzigen linearen Kopiervorgang.

Der Rettungsalgorithmus arbeitet in mehreren Phasen. Dabei werden zunächst möglichst große gut lesbare Bereiche kopiert. Problematische Regionen werden eingegrenzt und können anschließend mit feineren Zugriffen weiterbearbeitet werden.

Die GNU-ddrescue-Terminologie unterscheidet dabei insbesondere zwischen Phasen wie **copying**, **trimming**, **scraping** und **retrying**.

Der Zweck dieser Strategie besteht darin, zunächst möglichst viele problemlos erreichbare Daten zu sichern und erst danach mehr Zeit auf schwer lesbare Bereiche zu verwenden.

Erster Rettungsdurchlauf

Eine häufig verwendete Strategie besteht darin, im ersten Durchlauf zunächst auf zusätzliche Wiederholungsversuche für fehlerhafte Bereiche zu verzichten.

Ein entsprechender Aufruf kann beispielsweise lauten:

```
sudo ddrescue -f -n /dev/sdX /dev/sdY rescue.map
```

oder beim Schreiben in eine Image-Datei:

```
sudo ddrescue -n /dev/sdX rescue.img rescue.map
```

Die Option `-n` beziehungsweise `--no-scrape` überspringt die Scraping-Phase.

Das Ziel eines solchen ersten Durchlaufs besteht darin, zunächst möglichst viel leicht lesbares Material zu sichern, bevor zusätzliche Zeit und zusätzliche Lesezugriffe auf problematische Bereiche verwendet werden.

Weitere Rettungsversuche

Nach einem ersten Durchlauf können die noch nicht erfolgreich gelesenen Bereiche erneut bearbeitet werden.

Beispielsweise:

```
sudo ddrescue -d -r3 /dev/sdX rescue.img rescue.map
```

Die Option

`-r3`

begrenzt die Anzahl der Retry-Durchläufe auf drei.

`-r` beziehungsweise `--retry-passes` steuert die Anzahl der zusätzlichen Wiederholungsdurchläufe für als fehlerhaft markierte Bereiche.

Eine höhere Zahl ist nicht automatisch besser. Bei einem physisch instabilen Laufwerk bedeuten zusätzliche Versuche auch zusätzliche Belastung und Betriebszeit.

Direkter Datenträgerzugriff

Mit

`-d`

beziehungsweise

`--idirect`

kann ddrescue für die Eingabedatei direkten Datenträgerzugriff anfordern.

Diese Betriebsart kann bei bestimmten Rettungssituationen hilfreich sein, weil dabei der normale Kernel-Cache umgangen beziehungsweise dessen Einfluss reduziert wird.

Die tatsächliche Unterstützung hängt jedoch vom Betriebssystem und vom verwendeten Gerät ab.

Schreiben auf ein Blockgerät

Soll ddrescue direkt auf ein physisches Zielgerät schreiben, ist besondere Vorsicht erforderlich.

Die Option

`-f`

beziehungsweise

`--force`

ist erforderlich, wenn die Ausgabedatei ein reguläres Gerät beziehungsweise eine bereits vorhandene Gerätedatei ist, deren Überschreiben ddrescue andernfalls aus Sicherheitsgründen verweigert.

Beispiel:

```
sudo ddrescue -f /dev/sdX /dev/sdY rescue.map
```

Dieser Befehl kann den Inhalt des Zielgeräts überschreiben.

Deshalb müssen **Quelle und Ziel vor der Ausführung zweifelsfrei identifiziert werden**.

Image-Datei oder physischer Klon

GNU ddrescue kann eine Quelle sowohl auf einen anderen Datenträger als auch in eine Image-Datei kopieren.

Ein Image bietet für viele Rettungssituationen Vorteile.

Nach erfolgreicher Sicherung können Analysewerkzeuge auf dem Abbild arbeiten, während der ursprüngliche Datenträger nicht mehr für jede weitere Untersuchung benötigt wird.

In JUMBO BLOCK Rescue können anschließend beispielsweise TestDisk oder PhotoRec auf einem solchen Datenträgerabbild eingesetzt werden.

Ein physischer Klon kann dagegen sinnvoll sein, wenn ein unmittelbar als Datenträger nutzbares Duplikat benötigt wird.

Welche Variante verwendet wird, hängt vom Rettungsfall, dem verfügbaren Speicherplatz und dem Zustand des Quellmediums ab.

Zielmedium richtig dimensionieren

Beim direkten Klonen auf einen anderen Datenträger muss das Ziel ausreichend groß sein.

Bei einer Image-Datei muss auf dem Zieldateisystem entsprechend genügend freier Speicherplatz vorhanden sein.

Vor dem Start einer Rettung sollten deshalb sowohl die Größe des Quellgeräts als auch der verfügbare Speicherplatz des Zielmediums kontrolliert werden.

Dateisystemunabhängiges Arbeiten

GNU ddrescue arbeitet auf der Ebene der Eingabedaten beziehungsweise Blockgeräte.

Es benötigt daher für eine sektororientierte Rettung kein intaktes Dateisystem auf der Quelle.

Das ist ein wesentlicher Unterschied zu Werkzeugen, die für ihre Arbeit auf Dateisystemstrukturen angewiesen sind.

Ein beschädigtes NTFS-, FAT-, ext- oder anderes Dateisystem verhindert daher nicht grundsätzlich, dass die noch lesbaren Blöcke zunächst mit ddrescue in ein Abbild oder auf ein anderes Medium übertragen werden.

Die spätere Interpretation und Wiederherstellung der darin enthaltenen Partitionen und Dateien ist eine getrennte Aufgabe.

Datenträger möglichst nicht einhängen

Bei einer Datenrettung sollte vermieden werden, das beschädigte Quelldateisystem unnötig schreibend zu verwenden.

Insbesondere sollten keine Reparaturprogramme, automatischen Dateisystemreparaturen oder normalen Schreiboperationen auf dem Original ausgeführt werden, bevor eine Sicherung angelegt wurde, wenn die vorhandenen Daten von Bedeutung sind.

Für eine möglichst kontrollierte Rettung wird das Original daher bevorzugt als Quelle behandelt und die weitere Analyse auf dem erzeugten Klon beziehungsweise Image durchgeführt.

Keine Reparatur durch ddrescue

GNU ddrescue repariert weder Partitionstabellen noch Dateisysteme.

Es stellt auch keine gelöschten Dateien anhand ihrer Dateisysteminformationen oder Dateisignaturen wieder her.

Seine Aufgabe besteht zunächst darin, die noch lesbaren Daten von einer Quelle auf ein Ziel zu übertragen und den Rettungsfortschritt zu verwalten.

Die eigentliche logische Analyse erfolgt anschließend mit dafür vorgesehenen Werkzeugen.

In JUMBO BLOCK Rescue sind dies unter anderem:

TestDisk für Partitionierungs- und bestimmte Dateisystemstrukturen,

PhotoRec für signaturbasierte Dateirettung.

Umgang mit einem ausfallenden Datenträger

Ein physisch auffälliger Datenträger sollte möglichst wenig unnötig belastet werden.

Typische Warnzeichen können wiederkehrende I/O-Fehler, stark schwankende Leseraten oder vom Laufwerk beziehungsweise Betriebssystem gemeldete Hardwareprobleme sein.

Bei solchen Medien sollte die Priorität auf der Sicherung der noch lesbaren Daten liegen.

Jeder zusätzliche vollständige Scan und jeder unnötige Reparaturversuch auf dem Original verlängert die Betriebszeit des möglicherweise ausfallenden Geräts.

Deshalb ist in JUMBO BLOCK Rescue bei einem physisch problematischen Medium grundsätzlich die Reihenfolge sinnvoll:

Original identifizieren → Zielmedium vorbereiten → ddrescue mit Mapfile → Image/Klon sichern → weitere Analyse am Duplikat

Grenzen

GNU ddrescue kann nur Daten kopieren, die das Speichermedium und die darunterliegende Hardware noch zur Verfügung stellen können.

Ein Bereich, der auch nach wiederholten Leseversuchen physisch nicht mehr gelesen werden kann, kann durch Software nicht zuverlässig rekonstruiert werden.

Auch ein erfolgreicher ddrescue-Durchlauf bedeutet nicht automatisch, dass das enthaltene Dateisystem fehlerfrei oder jede Datei vollständig intakt ist.

Wenn einzelne Quellbereiche nicht gelesen werden konnten, fehlen diese Daten auch im erzeugten Abbild beziehungsweise Klon.

Das Ergebnis muss deshalb anschließend mit geeigneten Analyse- und Datenrettungswerkzeugen geprüft werden.

Sicherheitshinweise

GNU ddrescue arbeitet direkt mit Datenträgern und kann bei falscher Anwendung vorhandene Daten überschreiben.

Vor jedem Start müssen daher insbesondere folgende Punkte zweifelsfrei geklärt sein:

Welches Gerät ist die Quelle?

Welches Gerät beziehungsweise welche Datei ist das Ziel?

Ist das Ziel groß genug?

Ist ein vorhandenes Ziel überschreibbar?

Wird das richtige Mapfile verwendet?

Besonders kritisch ist die Verwechslung der Gerätenamen `/dev/sdX`, `/dev/sdY`, NVMe-Geräten oder anderer Blockgeräte.

Die in diesem Handbuch gezeigten Gerätenamen sind deshalb ausschließlich Beispiele und dürfen nicht ungeprüft übernommen werden.

Rolle in JUMBO BLOCK Rescue

GNU ddrescue ist innerhalb von JUMBO BLOCK Rescue das zentrale Werkzeug für die **Sicherung problematischer Datenträger vor der eigentlichen logischen Datenrettung**.

Es ergänzt TestDisk und PhotoRec, übernimmt jedoch eine klar andere Aufgabe.

Die drei Werkzeuge bilden zusammen eine abgestufte Rettungskette:

GNU ddrescue – möglichst viele noch lesbare Daten sichern

TestDisk – Partitionierungs- und unterstützte Dateisystemstrukturen untersuchen und gegebenenfalls wiederherstellen

PhotoRec – Dateien anhand ihrer Signaturen rekonstruieren, wenn die ursprünglichen Strukturen nicht mehr ausreichend nutzbar sind

Bei einem Datenträger mit Anzeichen physischer Probleme hat die Erstellung eines möglichst vollständigen Klons oder Images mit GNU ddrescue grundsätzlich Vorrang vor umfangreichen Reparaturversuchen am Original.

Das Original bleibt dabei die letzte verfügbare Quelle. Jede weitere Analyse sollte nach Möglichkeit auf dem gesicherten Abbild oder Klon erfolgen.

A.4 Foremost

Bereich: Datenrettung
Programm: Foremost
Baseline 20261001: Foremost 1.5.7
Installiertes Paket: `foremost 1.5.7-11+b2`
Bedienung: Kommandozeile im Terminal

Zweck

Foremost ist ein Werkzeug zur **signaturbasierten Dateiwiederherstellung beziehungsweise zum File Carving**. Es durchsucht Datenträgerabbilder oder andere Eingabedaten nach charakteristischen Headern, Footern und internen Datenstrukturen bekannter Dateiformate.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält Foremost als Paket `foremost` in Version `1.5.7-11+b2`.

Foremost wurde ursprünglich für forensische Untersuchungen entwickelt. Die Programmdokumentation beschreibt seinen Zweck als Wiederherstellung von Dateien anhand ihrer Header, Footer und Datenstrukturen.

Typische Einsatzfälle

Foremost eignet sich insbesondere für Situationen, in denen Dateien nicht mehr über eine intakte Verzeichnis- oder Dateisystemstruktur erreichbar sind.

Dazu gehören beispielsweise:

gelöschte Dateien,

beschädigte Dateisysteme,

Datenträgerabbilder aus einer vorherigen Rettung,

nicht mehr vollständig interpretierbare Partitionen,

sowie die gezielte Suche nach bestimmten Dateitypen innerhalb von Rohdaten.

Da Foremost anhand von Dateisignaturen arbeitet, ist für das eigentliche File Carving keine intakte ursprüngliche Verzeichnisstruktur erforderlich.

Arbeitsprinzip

Foremost sucht die Eingabedaten nach definierten Dateiköpfen – sogenannten **Headern** – und, soweit für den jeweiligen Dateityp vorgesehen, nach zugehörigen **Footern**.

Wird eine passende Struktur gefunden, versucht Foremost den entsprechenden Datenbereich als Datei zu extrahieren.

Die Regeln für die Erkennung können bereits im Programm definiert sein oder über eine Konfigurationsdatei bereitgestellt werden.

Damit arbeitet Foremost ähnlich wie PhotoRec auf der Ebene von Dateiinhalten und Signaturen und nicht primär über ursprüngliche Dateisystemeinträge.

Unterstützte integrierte Dateitypen

Die Foremost-1.5.7-Dokumentation führt eine Reihe vordefinierter Dateitypen auf.

Dazu gehören unter anderem:

`jpg` – JPEG/JFIF und Exif,

`gif`,

`png`,

`bmp`,

`avi`,

`exe` – Windows-PE-Dateien einschließlich DLL und EXE,

`mpg`,

`wav`,

`riff`,

`wmv`,

`mov`,

`pdf`,

`ole` – Dateien mit OLE-Struktur, darunter ältere Microsoft-Office-Formate,

`doc`,

`zip`,

`rar`,

`htm`,

`cpp`,

`mp4`.

Mit `all` können alle vordefinierten Extraktionsmethoden verwendet werden. Laut Manpage ist dies auch die Voreinstellung, wenn mit `-t` kein Dateityp angegeben wird.

Grundlegender Aufruf

Der grundlegende Programmaufruf lautet:

```
foremost [Optionen]
```

Eine Eingabedatei kann mit

`-i`

festgelegt werden.

Beispiel:

```
foremost -i rescue.img
```

Dabei wird `rescue.img` untersucht.

Wird keine Eingabedatei angegeben beziehungsweise kann die angegebene Eingabe nicht gelesen werden, verwendet Foremost laut Programmdokumentation die Standardeingabe `stdin`.

Auswahl des Ausgabeordners

Mit

`-o`

wird das Verzeichnis festgelegt, in das Foremost die wiederhergestellten Dateien schreibt.

Beispiel:

```
foremost -i rescue.img -o recovery
```

Damit wird `rescue.img` untersucht und das Ergebnis im Verzeichnis `recovery` abgelegt.

Quelle und Ausgabe sollten bei einer realen Datenrettung getrennt behandelt werden. Insbesondere dürfen wiederhergestellte Daten nicht unnötig auf einen beschädigten Originaldatenträger zurückgeschrieben werden.

Auswahl bestimmter Dateitypen

Mit

`-t`

kann die Suche auf bestimmte Dateitypen begrenzt werden.

Beispiel:

```
foremost -t jpg -i rescue.img -o recovery
```

sucht gezielt nach JPEG-Dateien.

Mehrere Typen können gemeinsam angegeben werden.

Die Foremost-Dokumentation zeigt beispielsweise:

```
foremost -t gif,pdf -i image.dd
```

für die Suche nach GIF- und PDF-Dateien.

Soll nach allen integrierten Typen gesucht werden, kann verwendet werden:

```
foremost -t all -i rescue.img
```

Verbose Mode

Die Option

`-v`

aktiviert den ausführlichen Ausgabemodus.

Foremost zeigt damit zusätzliche Informationen über den aktuellen Zustand des Programms an.

Die Programmdokumentation bezeichnet diesen Modus ausdrücklich als sehr empfehlenswert.

Beispiel:

```
foremost -v -i rescue.img -o recovery
```

Quick Mode

Mit

`-q`

wird der **Quick Mode** aktiviert.

In diesem Modus sucht Foremost nur am Anfang der einzelnen Sektoren nach passenden Headern. Dadurch kann die Verarbeitung erheblich schneller erfolgen.

Diese Geschwindigkeit wird jedoch mit einer eingeschränkten Suche erkauft.

Dateien, deren Header nicht an der erwarteten Position liegt, können übersehen werden. Die Foremost-Dokumentation nennt als Beispiel eingebettete JPEG-Bilder innerhalb von Microsoft-Word-Dokumenten.

Besonders wichtig ist der ausdrückliche Hinweis der Dokumentation:

Quick Mode sollte bei NTFS-Dateisystemen nicht verwendet werden.

NTFS kann kleine Dateien direkt innerhalb der Master File Table speichern. Solche Dateien können im Quick Mode übersehen werden.

Quiet Mode

Die Option

`-Q`

aktiviert den **Quiet Mode**.

Dabei werden die meisten Fehlermeldungen unterdrückt.

Für eine normale Rescue-Analyse ist diese Betriebsart nicht grundsätzlich vorteilhaft, da Fehlermeldungen wichtige Hinweise auf Probleme mit Quelle, Ausgabe oder Datenstrukturen enthalten können.

Audit-only-Modus

Mit

`-w`

kann Foremost im **Write-Audit-only-Modus** ausgeführt werden.

Dabei werden keine gefundenen Dateien extrahiert.

Diese Betriebsart ermöglicht damit eine Untersuchung beziehungsweise Protokollierung, ohne die gefundenen Dateien tatsächlich in das Ausgabeverzeichnis zu schreiben.

Option `-a`

Mit

`-a`

weist Foremost an, alle gefundenen Header zu verarbeiten und die normale Fehlererkennung hinsichtlich beschädigter Dateien nicht anzuwenden.

Diese Option kann dadurch zu Ergebnissen führen, die im normalen Betrieb aufgrund erkannter Inkonsistenzen nicht extrahiert würden.

Blockgröße

Mit

`-b`

kann die von Foremost verwendete Blockgröße festgelegt werden.

Die Standardeinstellung beträgt laut Foremost-Dokumentation **512 Byte**.

Beispiel:

```
foremost -b 1024 -i rescue.img
```

Die Blockgröße ist insbesondere für die Dateibenennung und die Suche im Quick Mode relevant.

Überspringen von Bereichen

Die Option

`-s`

ermöglicht es, eine bestimmte Anzahl von Blöcken am Anfang der Eingabe zu überspringen.

Die Dokumentation nennt beispielsweise:

```
foremost -s 100 -t jpg -i image.dd
```

Damit werden zunächst 100 Blöcke übersprungen, bevor die Suche nach JPEG-Dateien beginnt.

Diese Funktion kann für gezielte Untersuchungen bestimmter Bereiche eines Images verwendet werden.

Indirect Block Detection

Mit

`-d`

aktiviert Foremost die **indirect block detection**.

Laut Programmdokumentation funktioniert diese Option insbesondere gut mit Unix-Dateisystemen.

Sie kann beispielsweise zusammen mit der Suche nach bestimmten Dateitypen eingesetzt werden.

Ein von der Dokumentation gezeigtes Beispiel lautet sinngemäß:

```
foremost -vd -t ole,jpg -i image.dd
```

Damit werden OLE- und JPEG-Strukturen mit ausführlicher Ausgabe und aktivierter indirekter Blockerkennung untersucht.

Zeitgestempelte Ausgabeordner

Die Option

`-T`

ergänzt den Ausgabeordner um einen Zeitstempel.

Dadurch können mehrere Foremost-Läufe durchgeführt werden, ohne für jeden Lauf zuvor denselben Ausgabeordner entfernen beziehungsweise manuell einen neuen Namen vergeben zu müssen.

Dies ist insbesondere bei wiederholten Analysen desselben Images mit unterschiedlichen Einstellungen hilfreich.

Eigene Dateisignaturen

Eine wichtige Funktion von Foremost ist die Möglichkeit, die Erkennungsregeln über eine Konfigurationsdatei anzupassen.

Mit

`-c`

kann eine bestimmte Konfigurationsdatei angegeben werden.

Beispiel:

```
foremost -c eigene.conf -i rescue.img -o recovery
```

Wird keine Konfigurationsdatei explizit angegeben, sucht Foremost zunächst nach `foremost.conf` im aktuellen Verzeichnis und anschließend nach `/etc/foremost.conf`.

Aufbau von `foremost.conf`

Für einen Dateityp definiert die Konfigurationsdatei unter anderem:

die Dateierweiterung,

die Groß-/Kleinschreibung der Signaturprüfung,

die maximale Dateigröße,

den Header,

und optional den Footer.

Header, maximale Größe, Case-Sensitivity und Dateierweiterung sind erforderlich; der Footer ist optional.

Zeilen, die mit # beginnen, werden als Kommentare behandelt und ignoriert. Dadurch können einzelne Dateitypdefinitionen deaktiviert werden.

Binäre Signaturen

Foremost erlaubt die Definition von Signaturen auch in hexadezimaler beziehungsweise oktaler Schreibweise.

Hexadezimale Werte können beispielsweise mit

```
\xFF
```

angegeben werden.

Auch Wildcards sind innerhalb der Signaturdefinition möglich. Standardmäßig dient ? als Platzhalter für ein einzelnes beliebiges Zeichen.

Damit können eigene oder spezielle Dateiformate beschrieben werden, sofern ihre charakteristischen Datenstrukturen bekannt sind.

Audit-Datei

Foremost erzeugt bei seiner Arbeit eine Audit-Dokumentation des Durchlaufs.

Gerade aufgrund seiner Herkunft aus dem forensischen Umfeld ist die nachvollziehbare Dokumentation eines Carving-Laufs ein wichtiger Bestandteil des Programms.

Die Audit-Informationen sollten zusammen mit den geretteten Daten aufbewahrt werden, wenn die spätere Nachvollziehbarkeit des Rettungsvorgangs relevant ist.

Foremost und Datenträgerabbilder

Foremost ist besonders für die Untersuchung von Datenträgerabbildern geeignet.

Dies passt unmittelbar zur Rettungsstrategie von JUMBO BLOCK Rescue:

Ein physisch auffälliger Datenträger kann zunächst mit **GNU ddrescue** möglichst vollständig auf ein anderes Medium beziehungsweise in eine Image-Datei kopiert werden.

Anschließend kann Foremost das Image untersuchen, ohne für jeden Carving-Versuch erneut auf das problematische Originalmedium zugreifen zu müssen.

Beispiel:

```
foremost -v -i rescue.img -o foremost-recovery
```

Grenzen des File Carving

Foremost rekonstruiert Dateien anhand erkannter Datenstrukturen und Signaturen.

Das bedeutet nicht automatisch, dass jede gefundene Datei vollständig und fehlerfrei ist.

Insbesondere Fragmentierung kann ein Problem darstellen. Wenn die Bestandteile einer Datei nicht zusammenhängend gespeichert sind, kann eine rein signaturbasierte Rekonstruktion unvollständige oder fehlerhafte Ergebnisse erzeugen.

Ebenso können zufällige Bytefolgen einer bekannten Signatur entsprechen und dadurch zu falsch erkannten Ergebnissen führen.

Die extrahierten Dateien müssen deshalb anschließend geprüft werden.

Ursprüngliche Dateinamen und Verzeichnisstruktur

Foremost rekonstruiert grundsätzlich nicht die ursprüngliche Dateisystemorganisation.

Da die Dateien anhand ihres Inhalts beziehungsweise ihrer Signaturen gefunden werden, stehen ursprüngliche Dateinamen und Verzeichnispfade normalerweise nicht zur Verfügung.

Das ist ein grundlegender Unterschied zu einer erfolgreichen strukturbasierten Wiederherstellung mit TestDisk.

Foremost und PhotoRec

Foremost und PhotoRec gehören beide zur Gruppe der File-Carving-Werkzeuge.

Beide können Dateien anhand charakteristischer Dateiinhalte wiederfinden, ohne auf eine vollständig intakte ursprüngliche Dateisystemstruktur angewiesen zu sein.

Sie sind jedoch eigenständige Programme mit unterschiedlichen Erkennungsmechanismen, unterstützten Formaten, Konfigurationsmöglichkeiten und Bedienkonzepten.

Foremost bietet insbesondere eine explizite, editierbare Signaturkonfiguration über `foremost.conf`.

PhotoRec besitzt dagegen eine eigene umfangreiche Datenbank unterstützter Dateiformate und einen interaktiven textbasierten Arbeitsablauf.

Innerhalb von JUMBO BLOCK Rescue werden beide Werkzeuge deshalb bewusst separat bereitgestellt.

Sicherheitshinweise

Foremost sollte bei einem beschädigten Originaldatenträger nach Möglichkeit nicht als erster Schritt für umfangreiche wiederholte Scans verwendet werden.

Bei Anzeichen eines physischen Defekts sollte zunächst mit GNU ddrescue ein möglichst vollständiges Abbild beziehungsweise ein Klon erzeugt werden.

Danach kann Foremost auf diesem Abbild arbeiten.

Für die Ausgabe muss ausreichend Speicherplatz vorhanden sein. Eine breit angelegte Suche nach zahlreichen Dateitypen kann eine große Anzahl von Dateien erzeugen.

Das Ausgabeverzeichnis sollte außerdem auf einem sicheren Zielmedium liegen und nicht auf dem zu rettenden Quelldatenträger.

Rolle in JUMBO BLOCK Rescue

Foremost erweitert JUMBO BLOCK Rescue um ein zweites, eigenständiges Verfahren für die **signaturbasierte Dateirettung**.

Damit ergibt sich innerhalb des Bereichs Datenrettung:

TestDisk – Wiederherstellung und Reparatur unterstützter Partitionierungs- und Dateisystemstrukturen

PhotoRec – umfangreiches signaturbasiertes File Carving

GNU ddrescue – Sicherung problematischer Datenträger auf ein anderes Medium oder in ein Image

Foremost – header-/footer- und strukturorientiertes File Carving mit konfigurierbaren Signaturen

Foremost ist insbesondere dann interessant, wenn gezielt bestimmte Dateitypen gesucht werden sollen oder eigene Erkennungsregeln über `foremost.conf` erforderlich sind.

A.5 Extundelete

Bereich: Datenrettung
Programm: Extundelete
Baseline 20261001: Extundelete 0.2.4
Installiertes Paket: `extundelete 0.2.4-3+b2`
Bedienung: Kommandozeile im Terminal

Zweck

Extundelete ist ein spezialisiertes Datenrettungswerkzeug zur Wiederherstellung gelöschter Dateien von **ext3- und ext4-Dateisystemen**.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält das Paket `extundelete` in Version `0.2.4-3+b2`.

Extundelete nutzt Informationen aus dem Journal des Dateisystems, um zu versuchen, zuvor gelöschte Dateien wiederherzustellen. Die Programmdokumentation weist ausdrücklich darauf hin, dass die erfolgreiche Wiederherstellung einer bestimmten Datei nicht garantiert werden kann.

Einsatzbereich

Extundelete ist kein allgemeines File-Carving-Programm.

Sein Einsatzbereich ist ausdrücklich auf **ext3 und ext4** ausgerichtet. Es versucht, noch vorhandene Dateisystem- und Journalinformationen für die Wiederherstellung gelöschter Dateien zu verwenden.

Damit unterscheidet sich Extundelete von PhotoRec und Foremost, die Dateien primär anhand von Dateisignaturen suchen.

Wenn die benötigten Metadaten noch vorhanden sind, kann Extundelete Dateien unter ihrem ursprünglichen Namen beziehungsweise in Bezug auf ihre ursprünglichen Pfade wiederherstellen. Bei `--restore-all` werden Dateien, deren ursprünglicher Name nicht mehr bestimmt werden kann, unter einem Namen nach dem Muster `file.NNNN` ausgegeben.

Grundlegender Aufruf

Die Syntax lautet:

```
extundelete [Optionen] device-file
```

Als Quelle kann eine ext3- oder ext4-Partition angegeben werden.

Beispiel:

```
sudo extundelete /dev/sdXN --restore-all
```

Dabei sind `/dev/sdXN` und alle vergleichbaren Gerätenamen in diesem Handbuch ausschließlich Platzhalter. Das tatsächlich betroffene Gerät muss vor dem Einsatz zweifelsfrei identifiziert werden.

Extundelete kann außerdem mit einer **Datei arbeiten, die eine Kopie der Partition enthält**, beispielsweise einem zuvor erzeugten Rohabbild. Die Manpage nennt ausdrücklich eine mit `dd` erzeugte Kopie als mögliche Eingabe.

Damit lässt sich Extundelete auch auf einem zuvor gesicherten Partitionsabbild einsetzen.

Wiederherstellung einer einzelnen Datei

Mit

`--restore-file`

kann gezielt versucht werden, eine bestimmte gelöschte Datei wiederherzustellen.

Beispiel:

```
sudo extundelete /dev/sdXN --restore-file home/user/dokument.txt
```

Der Pfad wird relativ zur Wurzel des betreffenden Dateisystems angegeben.

Extundelete versucht anschließend, die angegebene Datei anhand der noch verfügbaren Informationen wiederherzustellen.

Wiederherstellung eines Verzeichnisses

Mit

`--restore-directory`

kann die Wiederherstellung der Dateien eines bestimmten Verzeichnisses versucht werden.

Beispiel:

```
sudo extundelete /dev/sdXN --restore-directory home/user/Documents
```

Die Manpage beschreibt diese Funktion als Versuch, alle Dateien wiederherzustellen, die dem angegebenen Verzeichnis zugeordnet werden können. Soweit möglich, werden dabei die Namen verwendet, die die Dateien vor ihrer Löschung besaßen.

Wiederherstellung aller auffindbaren Dateien

Mit

`--restore-all`

versucht Extundelete, alle Dateien wiederherzustellen, die sich mit den noch vorhandenen Informationen retten lassen.

Beispiel:

```
sudo extundelete /dev/sdXN --restore-all
```

Soweit die ursprünglichen Namen bestimmt werden können, werden diese verwendet. Für andere wiederhergestellte Dateien können Namen nach dem Schema

`file.NNNN`

erzeugt werden.

Diese Funktion kann bei einer größeren Löschaktion sinnvoll sein, erzeugt jedoch potenziell eine entsprechend umfangreiche Ergebnismenge.

Mehrere ausgewählte Dateien wiederherstellen

Mit

`--restore-files`

kann eine Liste von wiederherzustellenden Dateien verarbeitet werden.

Dazu wird zunächst eine Textdatei erstellt, die die gewünschten Pfade in derselben Form enthält, wie sie bei `--restore-file` angegeben würden.

Anschließend kann beispielsweise ausgeführt werden:

```
sudo extundelete /dev/sdXN --restore-files dateiliste.txt
```

Die Manpage weist darauf hin, dass diese Methode gegenüber vielen einzelnen Aufrufen Redundanz reduziert, weil das Journal nicht für jede einzelne Datei erneut vollständig verarbeitet werden muss.

Wiederherstellung anhand einer Inode-Nummer

Extundelete kann außerdem versuchen, Daten anhand bestimmter Inodes wiederherzustellen.

Dafür steht

`--restore-inode`

zur Verfügung.

Beispiel:

```
sudo extundelete /dev/sdXN --restore-inode 2569
```

Es können auch mehrere Inode-Nummern kommasepariert angegeben werden:

```
--restore-inode 2569,5692,6925
```

Diese Funktion ist insbesondere für eine gezielte technische Untersuchung interessant, wenn die relevanten Inodes bereits bekannt sind.

Superblock untersuchen

Mit

`--superblock`

kann Extundelete Informationen über den Superblock des Dateisystems ausgeben.

Beispiel:

```
sudo extundelete /dev/sdXN --superblock
```

Diese Funktion dient der Untersuchung des Dateisystems und führt nicht selbst die Wiederherstellung einer Datei durch.

Journal untersuchen

Die Kombination

```
--journal --superblock
```

gibt Informationen über das Journal anhand dessen Superblock aus.

Da Extundelete Informationen aus dem Journal für seine Wiederherstellungsversuche verwendet, ist dieses ein zentraler Bestandteil seines Arbeitsprinzips.

Inode-Informationen anzeigen

Mit

```
--inode
```

können Informationen zu einer bestimmten Inode-Nummer ausgegeben werden.

Beispiel:

```
sudo extundelete /dev/sdXN --inode 2
```

Dies ist eine Analysefunktion und kann bei der Untersuchung vorhandener Dateisystemstrukturen helfen.

Blockinhalt anzeigen

Mit

```
--block
```

kann der Inhalt eines bestimmten Blocks ausgegeben werden.

Beispiel:

```
sudo extundelete /dev/sdXN --block 9652
```

Auch hierbei handelt es sich um eine Analysefunktion für gezielte Untersuchungen des Dateisystems.

Zeitliche Eingrenzung

Extundelete kann Wiederherstellungsversuche anhand des Löschzeitpunkts zeitlich einschränken.

Dafür stehen

`--before`

und

`--after`

zur Verfügung.

Die Zeit wird als Anzahl der Sekunden seit der Unix-Epoche angegeben.

Mit

`--before`

werden Dateien berücksichtigt, die vor dem angegebenen Zeitpunkt gelöscht wurden.

Mit

`--after`

werden Dateien berücksichtigt, die nach dem angegebenen Zeitpunkt gelöscht wurden.

Ein Unix-Zeitwert kann unter Linux beispielsweise mit `date` erzeugt werden:

```
date -d "2026-10-01 10:00" +%s
```

Dadurch kann ein Wiederherstellungsversuch gezielt auf einen bekannten Zeitraum eingegrenzt werden.

Externes Journal

Ext-Dateisysteme können ein externes Journal verwenden.

Mit

`-j journal_dev`

kann Extundelete ein solches externes Journalgerät angegeben werden.

Diese Option ist nur relevant, wenn das zu untersuchende Dateisystem tatsächlich entsprechend konfiguriert wurde.

Backup-Superblock

Mit

`-b block_number`

kann ein bestimmter Backup-Superblock verwendet werden, wenn das Dateisystem geöffnet wird.

Zusätzlich kann mit

`-B block_size`

die zu verwendende Blockgröße angegeben werden.

Diese Optionen gehören zu den technisch fortgeschritteneren Funktionen und sollten nur eingesetzt werden, wenn Aufbau und Zustand des betroffenen Dateisystems ausreichend bekannt sind.

Warum schnelles Handeln wichtig ist

Extundelete kann eine gelöschte Datei nur dann erfolgreich wiederherstellen, wenn die dafür benötigten Informationen beziehungsweise Daten noch verfügbar sind.

Nach einer Löschung können freigegebene Datenblöcke vom Dateisystem erneut verwendet werden. Auch relevante Journalinformationen stehen nicht unbegrenzt zur Verfügung.

Deshalb sollten nach einer versehentlichen Löschung unnötige Schreibzugriffe auf das betroffene Dateisystem vermieden werden.

Die Programmdokumentation stellt ausdrücklich klar, dass eine erfolgreiche Wiederherstellung einer bestimmten Datei **nicht garantiert** werden kann.

Original oder Abbild

Für JUMBO BLOCK Rescue ist die Arbeit mit einem zuvor gesicherten Abbild besonders wichtig.

Die Extundelete-Manpage erlaubt als Eingabe ausdrücklich nicht nur eine Partition, sondern auch eine Datei, die eine Kopie dieser Partition enthält.

Damit kann ein sinnvoller Rescue-Ablauf lauten:

Originaldatenträger sichern → Partitionsabbild erzeugen → Extundelete auf dem Abbild einsetzen

Bei einem physisch auffälligen Datenträger kann GNU ddrescue für die vorherige Sicherung eingesetzt werden.

Das reduziert weitere Analysezugriffe auf das möglicherweise instabile Original.

Keine Garantie auf Wiederherstellung

Extundelete besitzt keine Möglichkeit, bereits physisch überschriebene Dateidaten wiederherzustellen.

Auch vorhandene Journalinformationen garantieren nicht, dass alle benötigten Datenblöcke einer gelöschten Datei noch verfügbar sind.

Die Manpage formuliert deshalb ausdrücklich, dass es **keine Garantie gibt, dass eine bestimmte Datei wiederhergestellt werden kann**.

Ein fehlgeschlagener Wiederherstellungsversuch bedeutet daher nicht zwangsläufig eine Fehlfunktion von Extundelete. Die für eine erfolgreiche Rekonstruktion notwendigen Informationen können schlicht nicht mehr vorhanden sein.

Abgrenzung zu TestDisk

TestDisk und Extundelete überschneiden sich teilweise im Bereich gelöschter Dateien, verfolgen jedoch unterschiedliche Schwerpunkte.

TestDisk ist ein umfassenderes Werkzeug zur Untersuchung und Wiederherstellung von Partitionierungs- und bestimmten Dateisystemstrukturen.

Extundelete ist dagegen gezielt auf die Wiederherstellung gelöschter Dateien aus **ext3- und ext4-Dateisystemen** ausgerichtet und verwendet dazu Informationen aus deren Journal.

Abgrenzung zu PhotoRec und Foremost

PhotoRec und Foremost arbeiten primär signaturbasiert.

Sie suchen charakteristische Datenstrukturen bekannter Dateiformate und können deshalb auch dann Ergebnisse liefern, wenn ursprüngliche Dateisysteminformationen nicht mehr ausreichend verfügbar sind.

Extundelete versucht dagegen, Informationen des ext3-/ext4-Dateisystems und seines Journals zu nutzen.

Der Vorteil einer erfolgreichen strukturbasierten Wiederherstellung besteht darin, dass ursprüngliche Namen und Zusammenhänge erhalten werden können.

Ist diese Wiederherstellung nicht mehr möglich, können PhotoRec oder Foremost als alternative File-Carving-Verfahren eingesetzt werden.

Abgrenzung zu GNU ddrescue

GNU ddrescue stellt keine gelöschten Dateien wieder her.

Seine Aufgabe besteht darin, noch lesbare Daten eines problematischen Datenträgers möglichst vollständig auf ein anderes Medium beziehungsweise in ein Abbild zu kopieren.

Extundelete kann anschließend auf einer solchen Kopie arbeiten, sofern darin ein geeignetes ext3- oder ext4-Dateisystem vorhanden ist.

Damit ergänzen sich beide Programme unmittelbar:

GNU ddrescue sichert die noch lesbaren Rohdaten.

Extundelete versucht anschließend, gelöschte Dateien aus ext3/ext4-Strukturen zu rekonstruieren.

Grenzen

Extundelete ist auf **ext3 und ext4** spezialisiert. Für NTFS, FAT, exFAT und andere Dateisysteme ist es nicht das vorgesehene Werkzeug.

Seine Wiederherstellungsmöglichkeiten hängen außerdem davon ab, welche Dateisystem-, Journal- und Dateidaten noch vorhanden sind.

Es ist deshalb kein Ersatz für ein Backup.

Auch der Hersteller beziehungsweise die Programmdokumentation weist ausdrücklich darauf hin, dass keine bestimmte Datei garantiert wiederhergestellt werden kann.

Sicherheitshinweise

Nach dem versehentlichen Löschen wichtiger Dateien sollte das betroffene ext3-/ext4-Dateisystem möglichst nicht weiter normal verwendet werden.

Neue Schreibvorgänge können freigegebene Datenblöcke erneut belegen und damit die Wiederherstellungschancen verschlechtern.

Wiederhergestellte Daten sollten deshalb auf einem **anderen Dateisystem beziehungsweise Zielmedium** abgelegt werden.

Bei Anzeichen eines Hardwaredefekts sollte außerdem nicht mit wiederholten intensiven Analyseversuchen am Original begonnen werden. In diesem Fall ist zunächst die Sicherung mit GNU ddrescue vorzuziehen.

Rolle in JUMBO BLOCK Rescue

Extundelete ergänzt die Datenrettungswerkzeuge von JUMBO BLOCK Rescue um ein spezialisiertes Verfahren für gelöschte Dateien auf Linux-Dateisystemen der Familien **ext3 und ext4**.

Damit umfasst der Bereich **Datenrettung** der Baseline 20261001 fünf unterschiedliche Werkzeuge beziehungsweise Verfahren:

TestDisk – Partitionierungs- und unterstützte Dateisystemstrukturen untersuchen und wiederherstellen.

PhotoRec – Dateien anhand ihrer Signaturen rekonstruieren.

GNU ddrescue – problematische Datenträger möglichst schonend sichern.

Foremost – konfigurierbares File Carving anhand von Headern, Footern und Datenstrukturen.

Extundelete – gelöschte Dateien aus ext3-/ext4-Dateisystemen unter Nutzung vorhandener Journalinformationen wiederherstellen.

Die Werkzeuge ersetzen sich damit nicht gegenseitig. Sie decken unterschiedliche Ebenen und Situationen einer Datenrettung ab und können innerhalb eines kontrollierten Rescue-Ablaufs aufeinander aufbauen.

A.6 GParted

Bereich: Datenträger
Programm: GParted
Baseline 20261001: GParted 1.6.0
Installierte Pakete: `gparted 1.6.0-2`, `gparted-common 1.6.0-2`
Bedienung: grafische Benutzeroberfläche

Zweck

GParted ist der **GNOME Partition Editor** zur grafischen Verwaltung von Partitionen und Partitionstabellen.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält GParted in Version `1.6.0-2`.

Das offizielle GParted-Handbuch für Version 1.6.0 beschreibt das Programm als Werkzeug zum Erstellen, Reorganisieren und Löschen von Partitionen. Zu den dokumentierten Funktionen gehören insbesondere das Erstellen von Partitionstabellen sowie das Erstellen, Löschen, Ändern der Größe, Verschieben, Prüfen, Bezeichnen, Kopieren und Einfügen von Partitionen.

Bedeutung im Rescue-System

GParted dient in JUMBO BLOCK Rescue zur grafischen Untersuchung und – wenn ausdrücklich erforderlich – Bearbeitung der Partitionierung angeschlossener Datenträger.

Das Programm kann beispielsweise verwendet werden, um vorhandene Datenträger und Partitionen zu identifizieren, deren Größe und Lage zu betrachten, nicht zugeordneten Speicherplatz zu erkennen und vorhandene Dateisysteme beziehungsweise Partitionseigenschaften zu untersuchen.

GParted ist jedoch **kein ausschließlich lesendes Diagnosewerkzeug**.

Viele seiner zentralen Funktionen verändern unmittelbar die Partitionierung oder die darauf befindlichen Dateisysteme. Deshalb muss zwischen der reinen Untersuchung eines Datenträgers und tatsächlich auszuführenden Änderungen strikt unterschieden werden.

Programmoberfläche

Nach dem Start zeigt GParted den ausgewählten Datenträger und dessen Partitionierung sowohl grafisch als auch tabellarisch an.

Über die Geräteauswahl kann zwischen erkannten Datenträgern gewechselt werden.

Die Darstellung ermöglicht eine schnelle Übersicht über vorhandene Partitionen und nicht zugeordneten Speicherplatz.

Vor jeder Bearbeitung muss kontrolliert werden, welcher physische Datenträger ausgewählt ist. Gerade in einem Rescue-System können gleichzeitig interne Laufwerke, USB-Datenträger und weitere Speichermedien angeschlossen sein.

Partitionen untersuchen

Eine vorhandene Partition kann ausgewählt und deren Informationen angezeigt werden.

Das offizielle Handbuch führt hierfür ausdrücklich die Funktion **„Informationen“** einer Partition auf.

Diese Untersuchung ist von Änderungen an der Partition zu unterscheiden.

Solange keine verändernde Operation ausgeführt wird, kann GParted daher zunächst zur Orientierung über die vorhandene Datenträgerstruktur dienen.

Unterstützte Operationen

GParted 1.6.0 unterstützt abhängig vom jeweiligen Dateisystem und den installierten Hilfsprogrammen unterschiedliche Operationen.

Zu den grundsätzlich dokumentierten Partitionsoperationen gehören:

Erstellen

Löschen

Größe ändern

Verschieben

Prüfen

Dateisystembezeichnung setzen

Kopieren

Einfügen

Partitionsmarkierungen verwalten

Darüber hinaus kann GParted Partitionstabellen auf Datenträgern erzeugen.

Nicht jede Operation steht für jedes Dateisystem gleichermaßen zur Verfügung. Die tatsächlichen Möglichkeiten hängen vom Dateisystem und den auf dem System verfügbaren Werkzeugen ab.

Partitionstabelle erstellen

GParted kann eine neue Partitionstabelle auf einem Datenträger erzeugen.

Dies ist eine **destruktive beziehungsweise potenziell datenvernichtende Operation**.

Eine neue Partitionstabelle darf auf einem Datenträger mit zu rettenden Daten nicht leichtfertig angelegt werden.

Insbesondere ist das Erstellen einer neuen Partitionstabelle **keine Reparaturmaßnahme für eine lediglich beschädigte oder verlorene vorhandene Partitionstabelle**.

Wenn eine vorhandene Partitionierungsstruktur wiederhergestellt werden soll, ist innerhalb von JUMBO BLOCK Rescue zunächst ein dafür vorgesehenes Werkzeug wie TestDisk zu prüfen.

Neue Partition erstellen

Nicht zugeordneter Speicherplatz kann zum Erstellen einer neuen Partition verwendet werden.

Dabei können abhängig von der gewählten Partitionstabelle und den verfügbaren Dateisystemwerkzeugen verschiedene Eigenschaften festgelegt werden.

Das Erstellen einer neuen Partition verändert die Datenträgerstruktur.

Auf einem Rescue-Original mit verlorenen Daten sollte deshalb nicht ohne vorherige Sicherung neuer Speicherplatz partitioniert oder formatiert werden.

Partition löschen

GParted kann vorhandene Partitionen löschen.

Das Löschen entfernt den entsprechenden Partitionseintrag aus der vorgesehenen Datenträgerstruktur.

Diese Funktion ist im normalen Administrationsbetrieb sinnvoll, stellt im Datenrettungsfall jedoch eine schreibende Veränderung am Originalmedium dar.

Eine versehentlich gelöschte oder beschädigte Partition sollte daher nicht durch weitere experimentelle Änderungen an der Partitionstabelle „repariert“ werden.

Partition formatieren

GParted kann Partitionen mit unterstützten Dateisystemen formatieren, sofern die dafür benötigten Werkzeuge vorhanden sind.

Eine Formatierung erzeugt neue Dateisystemstrukturen und kann vorhandene Datenstrukturen überschreiben.

Eine Partition mit zu rettenden Daten darf deshalb nicht formatiert werden.

Das gilt auch dann, wenn ein Betriebssystem meldet, eine Partition müsse formatiert werden, bevor sie verwendet werden könne.

Eine solche Meldung ist kein Beleg dafür, dass keine rettbaren Daten mehr vorhanden sind.

Größe einer Partition ändern

GParted kann die Größe unterstützter Partitionen und Dateisysteme verändern.

Hierfür wird **Partition → Größe ändern/Verschieben** verwendet.

Nicht eingehängte beziehungsweise inaktive Partitionen bieten laut GParted-Handbuch die meisten Möglichkeiten zur Größenänderung. Für einige aktive Dateisysteme ist eine Online-Größenänderung möglich, diese ist jedoch häufig auf eine Vergrößerung beschränkt.

Für eine Vergrößerung muss geeigneter nicht zugeteilter Speicherplatz vorhanden sein.

Partition verschieben

GParted kann eine Partition innerhalb des verfügbaren Speicherbereichs verschieben.

Auch dies erfolgt über **Größe ändern/Verschieben**.

Ein Verschieben kann erheblich länger dauern als eine reine Änderung von Partitionsgrenzen, da Daten physisch beziehungsweise logisch an andere Positionen übertragen werden müssen.

Das offizielle Handbuch warnt außerdem ausdrücklich davor, dass das Verschieben einer Betriebssystem-Bootpartition dazu führen kann, dass das Betriebssystem anschließend nicht mehr startet.

Eine solche Operation darf daher nicht als unverbindlicher Test betrachtet werden.

Partition kopieren und einfügen

GParted kann eine nicht eingehängte Partition kopieren und in geeigneten nicht zugeordneten Speicherplatz einfügen.

Dabei wird die Quellpartition zunächst mit **Partition → Kopieren** ausgewählt und anschließend über **Partition → Einfügen** in den Zielbereich übertragen.

Das offizielle Handbuch weist darauf hin, dass die Kopie zunächst dieselbe Dateisystembezeichnung und UUID wie die Quellpartition besitzt.

Dies muss berücksichtigt werden, wenn Original und Kopie gleichzeitig in einem System verwendet werden.

Partition prüfen

Für unterstützte Dateisysteme stellt GParted eine Funktion **Partition → Prüfen** bereit.

Dabei werden die jeweils verfügbaren Dateisystemwerkzeuge verwendet.

Eine solche Prüfung ist nicht mit einer rein passiven Anzeige gleichzusetzen. Abhängig vom Dateisystem und eingesetzten Werkzeug können Reparaturmaßnahmen Bestandteil der Operation sein.

Bei einem Datenträger mit wichtigen zu rettenden Daten sollte daher vor einer Dateisystemreparatur zunächst eine Sicherung beziehungsweise ein Image des Originals erstellt werden.

Partitionsmarkierungen

GParted kann Partitionsflags aktivieren und deaktivieren.

Das offizielle Handbuch nennt beispielsweise die Markierungen `boot` und `hidden`.

Welche Flags tatsächlich verfügbar sind, hängt von der verwendeten Partitionstabelle und der ausgewählten Partition ab.

Das Ändern solcher Flags verändert Partitionsmetadaten und sollte deshalb ebenfalls bewusst erfolgen.

Ausstehende Operationen

Eine besonders wichtige Eigenschaft von GParted ist die Verwaltung **ausstehender Operationen**.

Viele Änderungen werden zunächst nicht unmittelbar ausgeführt, sondern in einer Warteschlange vorgemerkt.

Dadurch können geplante Operationen vor der tatsächlichen Ausführung nochmals kontrolliert werden.

Die Warteschlange kann korrigiert werden, solange die Änderungen noch nicht angewendet wurden.

Das offizielle Handbuch beschreibt Funktionen zum Rückgängigmachen der letzten Operation und zum Löschen aller ausstehenden Operationen. Erst mit **„Alle Operationen ausführen“** werden die vorgemerkten Änderungen tatsächlich abgearbeitet.

Vor dem Anwenden kontrollieren

Der entscheidende Sicherheitsmoment ist das Anwenden der vorgemerkten Operationen.

Vor diesem Schritt müssen insbesondere kontrolliert werden:

der ausgewählte Datenträger,

die betroffene Partition,

die geplante Operation,

Größe und Position,

das Ziel einer Kopieroperation,

sowie die Reihenfolge mehrerer ausstehender Operationen.

GParted warnt selbst ausdrücklich davor, dass das Bearbeiten von Partitionen zu **Datenverlust** führen kann und empfiehlt eine Datensicherung vor der Verwendung.

Eingehängte Partitionen

Viele Partitionsoperationen erfordern eine nicht eingehängte beziehungsweise inaktive Partition.

Dies ist einer der Gründe, warum ein extern gestartetes Rescue-System für Partitionsarbeiten nützlich ist: Die Systempartitionen des installierten Betriebssystems müssen dabei nicht zwangsläufig aktiv verwendet werden.

GParted 1.6.0 unterstützt für bestimmte Dateisysteme zwar einzelne Operationen auch im eingehängten Zustand, die Möglichkeiten sind jedoch eingeschränkt.

Verschlüsselte Partitionen

GParted besitzt Unterstützung für bestimmte Operationen mit verschlüsselten Partitionen.

Für LUKS gilt beispielsweise laut offiziellem Handbuch, dass eine verschlüsselte Partition nur verschoben werden kann, wenn das Verschlüsselungs-Mapping geschlossen ist.

Bei verschlüsselten Datenträgern ist besondere Vorsicht erforderlich.

Das GParted-Handbuch weist ausdrücklich darauf hin, dass vor Partitionsänderungen eine Datensicherung empfohlen wird und hebt dies besonders für verschlüsselte Daten hervor.

Dateisystemunterstützung

GParted unterstützt zahlreiche Dateisysteme, jedoch nicht jede Operation für jedes Dateisystem.

Die offizielle Funktionsübersicht unterscheidet deshalb für jedes Dateisystem einzeln zwischen Funktionen wie:

Erkennen,

Lesen,

Erstellen,

Vergrößern,

Verkleinern,

Verschieben,

Kopieren,

Prüfen,

Bezeichnung ändern,

UUID ändern.

Im Handbuch von JUMBO BLOCK Rescue sollte daher nicht pauschal davon ausgegangen werden, dass jede GParted-Funktion für jedes erkannte Dateisystem verfügbar ist.

Entscheidend ist die im konkreten System von GParted angezeigte Verfügbarkeit.

Datenträgertypen

GParted kann laut Projekt-Dokumentation mit unterschiedlichen Speichermedien arbeiten.

Dazu gehören klassische Festplatten sowie Flash-Speicher wie USB-Sticks und SSDs. Die GParted-Dokumentation nennt außerdem NVMe-Geräte und verschiedene RAID-Konfigurationen.

Für den Rescue-Einsatz bleibt unabhängig vom Gerätetyp entscheidend, Quelle und Ziel eindeutig zu identifizieren.

Besonderheit von GParted 1.6.0

Für die in JUMBO BLOCK Rescue verwendete Version 1.6.0 ist eine Änderung gegenüber älteren GParted-Versionen besonders wichtig:

Mit GParted **1.6.0 wurde die frühere Funktion „Attempt Data Rescue“ entfernt**, ebenso die Verwendung von `gpart` für diese Funktion. Dies wird in der offiziellen Veröffentlichung zu GParted 1.6.0 ausdrücklich genannt.

Das bedeutet für dieses Handbuch:

GParted 1.6.0 wird nicht als Werkzeug zur Wiederherstellung verlorener Partitionen dokumentiert.

Für die eigentliche Partitionswiederherstellung stehen in JUMBO BLOCK Rescue andere Werkzeuge zur Verfügung, insbesondere TestDisk.

GParted und TestDisk

Die Aufgaben von GParted und TestDisk müssen im Rescue-Betrieb klar getrennt werden.

GParted dient der Anzeige und bewussten Verwaltung vorhandener Partitionierungsstrukturen und Dateisysteme.

TestDisk ist auf die Untersuchung und Wiederherstellung verlorener beziehungsweise beschädigter Partitionierungs- und bestimmter Dateisystemstrukturen ausgerichtet.

Wenn eine Partition lediglich verloren gegangen oder die Partitionstabelle beschädigt ist, sollte deshalb nicht vorschnell mit GParted eine neue Partition oder Partitionstabelle erzeugt werden.

Dadurch könnten Strukturen verändert werden, die TestDisk noch für eine Wiederherstellung verwenden könnte.

GParted und GNU ddrescue

Auch GParted und GNU ddrescue erfüllen unterschiedliche Aufgaben.

GParted verwaltet Partitionen und Partitionstabellen.

GNU ddrescue sichert möglichst viele noch lesbare Rohdaten eines problematischen Datenträgers.

Bei einem physisch auffälligen Datenträger sollte daher nicht zunächst mit umfangreichen GParted-Operationen begonnen werden.

Stattdessen ist es sinnvoll, zunächst ein möglichst vollständiges Image beziehungsweise einen Klon mit GNU ddrescue anzulegen und weitere Untersuchungen möglichst am Duplikat durchzuführen.

Typischer Rescue-Ablauf

Bei einem unbekannten oder beschädigten Datenträger kann GParted zunächst zur visuellen Orientierung verwendet werden.

Dabei wird der Datenträger ausgewählt und die angezeigte Partitionierung geprüft.

Sind die vorhandenen Strukturen plausibel und besteht lediglich administrativer Änderungsbedarf, können geeignete GParted-Operationen geplant werden.

Besteht dagegen der Verdacht auf verlorene Partitionen, beschädigte Dateisystemstrukturen oder einen Hardwaredefekt, sollte **nicht vorschnell schreibend eingegriffen werden**.

Dann sind zunächst die spezialisierten Datenrettungs- und Diagnosewerkzeuge von JUMBO BLOCK Rescue einzusetzen.

Grenzen

GParted ist kein Ersatz für ein Backup.

Es ist kein Werkzeug zur physischen Datenrettung von ausfallenden Datenträgern.

Und GParted 1.6.0 besitzt nicht mehr die frühere Funktion „Attempt Data Rescue“.

Die erfolgreiche Durchführung einer Partitionsoperation garantiert außerdem nicht, dass ein beschädigtes Betriebssystem anschließend wieder startet oder ein zuvor beschädigtes Dateisystem vollständig fehlerfrei ist.

Sicherheitshinweise

Das GParted-Projekt warnt ausdrücklich:

Das Bearbeiten von Partitionen kann zu Datenverlust führen. Softwarefehler, Hardwareprobleme oder ein Stromausfall während einer Operation können ebenfalls Datenverlust verursachen. Vor Partitionsänderungen wird deshalb eine Datensicherung empfohlen.

Für JUMBO BLOCK Rescue gilt daher:

Erst identifizieren, dann handeln.

Vor jeder schreibenden Operation müssen Datenträger und Partition eindeutig bestimmt sein.

Bei wertvollen oder nicht anderweitig gesicherten Daten sollte vor strukturellen Änderungen möglichst ein Backup, Image oder Klon erstellt werden.

Bei physisch auffälligen Medien hat die Sicherung mit GNU ddrescue Vorrang.

Bei verlorenen Partitionen sollte zunächst TestDisk eingesetzt werden, bevor neue Partitionsstrukturen mit GParted geschrieben werden.

Rolle in JUMBO BLOCK Rescue

GParted bildet innerhalb von JUMBO BLOCK Rescue das zentrale **grafische Werkzeug zur Anzeige und Verwaltung von Partitionen und Partitionstabellen**.

Es ergänzt die bisher dokumentierten Datenrettungsprogramme um die administrative Ebene der Datenträgerstruktur.

Dabei gilt eine klare funktionale Trennung:

GNU ddrescue – Datenträger sichern

TestDisk – verlorene beziehungsweise beschädigte Partitionierungsstrukturen untersuchen und wiederherstellen

PhotoRec und Foremost – Dateien signaturbasiert retten

Extundelete – gelöschte Dateien aus ext3/ext4 wiederherstellen

GParted – vorhandene Partitionen und Partitionstabellen grafisch anzeigen und bewusst verwalten

Gerade weil GParted leistungsfähige schreibende Funktionen besitzt, ist es im Rescue-System sowohl ein wichtiges Werkzeug als auch ein Werkzeug, das mit besonderer Sorgfalt eingesetzt werden muss.

A.7 Laufwerke / GNOME Disks

Bereich: Datenträger
Programm: Laufwerke / GNOME Disks
Baseline 20261001: GNOME Disks 46.1
Installiertes Paket: `gnome-disk-utility 46.1-2+b1`
Bedienung: grafische Benutzeroberfläche

Zweck

Laufwerke, international als **GNOME Disks** bezeichnet, ist das grafische Datenträgerverwaltungswerkzeug des GNOME-Projekts.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält `gnome-disk-utility` in Version `46.1-2+b1`.

GNOME beschreibt die Anwendung als Werkzeug zum **Inspizieren, Formatieren, Partitionieren und Konfigurieren von Datenträgern und Blockgeräten**. Darüber hinaus können SMART-Daten angezeigt, physische Laufwerke einem Leistungstest unterzogen und Datenträgerabbilder verwendet werden.

Damit verbindet GNOME Disks mehrere Funktionen, die bei einem Rescue-System relevant sind:

Datenträger identifizieren,

Partitionen und Dateisysteme untersuchen,

Laufwerkszustand kontrollieren,

SMART-Daten anzeigen und Selbsttests starten,

Laufwerksleistung messen,

Datenträgerabbilder verwenden,

Partitionen verwalten,

Dateisysteme prüfen und reparieren,

sowie Datenträger beziehungsweise Partitionen formatieren.

Ein Teil dieser Funktionen ist rein diagnostisch, andere Funktionen verändern den Datenträger. Diese Unterscheidung ist im Rescue-Betrieb entscheidend.

Datenträgerübersicht

Nach dem Start zeigt GNOME Disks die erkannten Speichergeräte in einer Geräteliste an.

Nach Auswahl eines Geräts werden Informationen zum jeweiligen Datenträger und dessen vorhandenen Volumes beziehungsweise Partitionen dargestellt.

Dadurch kann das Programm zunächst zur **Identifikation eines angeschlossenen Datenträgers** verwendet werden.

Dies ist vor jeder weiteren Rescue-Maßnahme wichtig.

Bei mehreren angeschlossenen Laufwerken muss eindeutig festgestellt werden, welches Gerät Quelle und welches gegebenenfalls Ziel einer Operation ist.

Datenträger und Blockgeräte

GNOME Disks arbeitet nicht ausschließlich mit klassischen Festplatten.

Die offizielle GNOME-Beschreibung spricht allgemein von **Disks and Block Devices**. Damit ist die Anwendung als Verwaltungswerkzeug für vom Linux-System erkannte entsprechende Speicher- und Blockgeräte konzipiert.

Im Rescue-Betrieb können deshalb beispielsweise interne und externe Laufwerke sowie USB-Speichermedien über die grafische Oberfläche untersucht werden, sofern sie vom System als entsprechende Geräte erkannt werden.

Partitionen und Volumes

GNOME Disks stellt die vorhandene Struktur eines ausgewählten Datenträgers grafisch dar.

Damit lassen sich vorhandene Partitionen beziehungsweise Volumes voneinander unterscheiden und grundlegende Eigenschaften der jeweiligen Bereiche untersuchen.

Die GNOME-Dokumentation führt ausdrücklich die Verwaltung von **Volumes und Partitionen** als Aufgabe der Laufwerksverwaltung auf.

Dies macht GNOME Disks zu einem geeigneten ersten grafischen Werkzeug, um sich einen Überblick über einen unbekannten Datenträger zu verschaffen.

Einhängen und Aushängen

Dateisysteme beziehungsweise Volumes können über die grafische Oberfläche ein- und ausgehängt werden, sofern das jeweilige Gerät und Dateisystem dies zulassen.

Für administrative Operationen kann es erforderlich sein, ein Dateisystem zunächst auszuhängen.

Im Rescue-Betrieb ist zusätzlich zu beachten, dass das Einhängen eines beschädigten oder zu sichernden Dateisystems nicht automatisch der erste sinnvolle Schritt sein muss.

Wenn ein Hardwaredefekt vermutet wird, sollte zunächst geprüft werden, ob eine Sicherung des Mediums mit GNU ddrescue vorzuziehen ist.

SMART-Daten

GNOME Disks kann bei unterstützten Laufwerken **SMART-Daten** anzeigen.

SMART steht für **Self-Monitoring, Analysis and Reporting Technology**.

GNOME beschreibt SMART als eine in Laufwerken integrierte Technik zur Überwachung möglicher Probleme. In GNOME Disks kann über **SMART Data & Self-Tests** die allgemeine Beurteilung des Laufwerks eingesehen und auf einzelne SMART-Attribute zugegriffen werden.

Damit ermöglicht GNOME Disks eine grafische erste Zustandsbeurteilung eines unterstützten Laufwerks.

SMART-Selbsttests

Neben der Anzeige vorhandener SMART-Daten können über GNOME Disks SMART-Selbsttests gestartet werden.

Die GNOME-Dokumentation beschreibt dazu den Aufruf von **SMART Data & Self-Tests** und anschließend **Start Self-test**.

Das Ergebnis ergänzt die bereits vorhandenen SMART-Attribute und die allgemeine Laufwerksbeurteilung.

SMART ist jedoch keine Garantie dafür, dass ein Laufwerk fehlerfrei ist oder nicht kurzfristig ausfallen kann.

Ein auffälliger Zustand sollte insbesondere bei wichtigen Daten als Anlass verstanden werden, die Daten möglichst schnell zu sichern.

Abgrenzung zu A.8 – SMART Diagnose

Die SMART-Funktion von GNOME Disks wird in diesem Kapitel nur als Bestandteil der grafischen Laufwerksverwaltung behandelt.

JUMBO BLOCK Rescue enthält zusätzlich den eigenständigen Menüpunkt beziehungsweise Handbuchabschnitt **A.8 – SMART Diagnose**.

Dort wird die eigentliche SMART-Diagnostik mit dem in der Baseline vorhandenen `smartmontools`-Werkzeugsatz separat und wesentlich detaillierter dokumentiert.

Damit bleibt die Trennung eindeutig:

GNOME Disks – grafische Laufwerksverwaltung einschließlich SMART-Anzeige

SMART Diagnose – spezialisierte Laufwerksdiagnostik

Leistungstest

GNOME Disks besitzt einen integrierten **Laufwerksbenchmark**.

Nach Auswahl des Laufwerks kann über das Laufwerksmenü der Leistungstest gestartet werden. Dabei können Parameter für Übertragungsrate und Zugriffszeit angepasst werden.

Ein reiner Lesetest ermittelt die Geschwindigkeit, mit der Daten vom Medium gelesen werden.

Nach Abschluss zeigt GNOME Disks die Messergebnisse grafisch an.

Dokumentiert werden unter anderem:

minimale Übertragungsrate,

maximale Übertragungsrate,

durchschnittliche Übertragungsrate,

sowie durchschnittliche Zugriffszeit.

Der Benchmark kann damit zur technischen Beurteilung der Laufwerksleistung verwendet werden.

Schreibbenchmark

Der Benchmark besitzt zusätzlich eine Option zum **Schreibtest**.

Wird diese aktiviert, misst GNOME Disks neben der Lese- auch die Schreibgeschwindigkeit.

Für ein Rescue-System ist diese Unterscheidung wesentlich:

Ein Lesetest und ein Schreibtest sind sicherheitstechnisch nicht gleichzusetzen.

Ein Schreibbenchmark erzeugt Schreibzugriffe auf das Medium und sollte deshalb nicht unüberlegt auf einem Originaldatenträger mit zu rettenden Daten eingesetzt werden.

Bei einem beschädigten oder forensisch beziehungsweise technisch unverändert zu haltenden Quellmedium ist ein Schreibbenchmark nicht als unverbindliche Diagnose geeignet.

Datenträgerabbilder

GNOME Disks unterstützt die Arbeit mit **Datenträgerabbildern**.

Das GNOME-Projekt nennt das Erstellen beziehungsweise Verwenden von Abbildern von USB-Datenträgern ausdrücklich als Funktion der Anwendung.

Damit können Daten eines geeigneten Datenträgers in eine Abbilddatei übertragen beziehungsweise vorhandene Abbilder wieder auf einen Datenträger geschrieben werden.

Die beiden Richtungen müssen strikt unterschieden werden:

Datenträger → Abbilddatei

und

Abbilddatei → Datenträger

Die zweite Richtung überschreibt Daten auf dem Zielmedium.

Datenträgerabbild erstellen

Das Erstellen eines Datenträgerabbilds kann dazu verwendet werden, den Inhalt eines Datenträgers in einer Image-Datei zu sichern.

Für einen intakten oder problemlos lesbaren Datenträger kann dies eine praktische grafische Funktion sein.

Ein solches Abbild kann anschließend getrennt vom ursprünglichen Medium aufbewahrt oder weiter untersucht werden.

Datenträgerabbild wiederherstellen

GNOME Disks kann ein vorhandenes Datenträgerabbild wieder auf ein ausgewähltes Laufwerk schreiben.

Diese Operation ist **destruktiv für den bisherigen Inhalt des Zielgeräts**, weil das Ziel mit den Daten des Images beschrieben wird.

Vor dem Start müssen daher mindestens drei Dinge eindeutig kontrolliert werden:

die ausgewählte Image-Datei,

der ausgewählte Zieldatenträger,

und die Tatsache, dass vorhandene Daten auf diesem Ziel überschrieben werden dürfen.

Eine Verwechslung von Quell- und Zieldatenträger kann zu erheblichem Datenverlust führen.

Abgrenzung zu GNU ddrescue

Die Image-Funktion von GNOME Disks darf nicht mit GNU ddrescue gleichgesetzt werden.

GNOME Disks stellt eine komfortable grafische Möglichkeit zur Arbeit mit Datenträgerabbildern bereit.

GNU ddrescue ist dagegen speziell für die Datenrettung von problematisch lesbaren Medien konzipiert und arbeitet mit einem Mapfile, dokumentiert bereits gerettete beziehungsweise problematische Bereiche und ermöglicht darauf aufbauende weitere Rettungsdurchläufe.

Bei einem **physisch auffälligen, instabilen oder fehlerhaft lesbaren Datenträger** ist deshalb GNU ddrescue das dafür spezialisierte Werkzeug innerhalb von JUMBO BLOCK Rescue.

Partitionen erstellen und verwalten

GNOME Disks kann Partitionen beziehungsweise Volumes verwalten und Datenträger partitionieren. Das GNOME-Projekt nennt Partitionierung ausdrücklich als Kernfunktion der Anwendung.

Solche Operationen verändern die Datenträgerstruktur.

Im Rescue-Betrieb sollten deshalb nicht ohne klare technische Begründung neue Partitionen auf einem Medium mit verlorenen Daten angelegt werden.

Ist eine Partition lediglich verschwunden oder eine Partitionierungsstruktur beschädigt, sollte zunächst ein spezialisiertes Wiederherstellungswerkzeug wie TestDisk eingesetzt werden.

Formatieren

GNOME Disks kann Partitionen beziehungsweise geeignete Datenträger formatieren.

GNOME beschreibt die Formatierungsfunktion ausdrücklich als Möglichkeit, vorhandene Dateien eines Wechseldatenträgers zu entfernen und ein Dateisystem für die weitere Verwendung einzurichten.

Für die Datenrettung gilt daher:

Eine Partition mit zu rettenden Daten darf nicht formatiert werden, nur weil das Betriebssystem das vorhandene Dateisystem nicht mehr korrekt erkennt.

Formatieren erzeugt neue Dateisystemstrukturen und verändert damit das Medium.

Formatieren ist kein sicheres Löschen

GNOME weist ausdrücklich darauf hin, dass normales Formatieren **kein Verfahren zur sicheren Datenlöschung** darstellt.

Nach einer Formatierung können mit speziellen Wiederherstellungswerkzeugen unter Umständen weiterhin Dateien beziehungsweise Daten gefunden werden.

Diese Eigenschaft ist aus Rescue-Sicht relevant:

Eine versehentliche Formatierung bedeutet nicht zwangsläufig, dass sämtliche Daten unwiederbringlich verloren sind.

In diesem Fall sollten weitere Schreibzugriffe vermieden und geeignete Datenrettungswerkzeuge eingesetzt werden.

Dateisystemgröße verändern

Die GNOME-Laufwerksverwaltung unterstützt bei geeigneten Dateisystemen die Veränderung der Größe eines Dateisystems und seiner Partition.

Die offizielle GNOME-Dokumentation führt ausdrücklich das **Verkleinern und Vergrößern eines Dateisystems und seiner Partition** als Funktion der Datenträgerverwaltung auf.

Wie bei GParted handelt es sich hierbei um eine strukturelle Veränderung des Datenträgers.

Eine solche Operation sollte auf einem beschädigten Originalmedium nicht als erster Reparaturversuch eingesetzt werden.

Dateisystem prüfen und reparieren

GNOME Disks beziehungsweise die GNOME-Datenträgerverwaltung bietet Funktionen zum Prüfen und Reparieren unterstützter Dateisysteme.

Die GNOME-Dokumentation führt ausdrücklich die Aufgabe **„Repair a damaged filesystem“** auf.

Eine Dateisystemreparatur ist von einer rein lesenden Diagnose zu unterscheiden.

Reparaturwerkzeuge können Dateisystemstrukturen verändern, um wieder einen konsistenten Zustand herzustellen.

Bei wichtigen Daten sollte deshalb möglichst zunächst ein Image oder eine andere Sicherung des Ausgangszustands angelegt werden.

Laufwerk ausschalten

Bei unterstützten Geräten kann GNOME Disks Funktionen zum sicheren Beenden beziehungsweise Ausschalten eines Laufwerks bereitstellen.

Dies ist insbesondere bei extern angeschlossenen Speichermedien nützlich, die nach Abschluss der Arbeiten kontrolliert vom System getrennt werden sollen.

Vor dem physischen Entfernen eines Mediums müssen laufende Schreibvorgänge abgeschlossen und verwendete Dateisysteme ordnungsgemäß ausgehängt sein.

GNOME Disks und GParted

GNOME Disks und GParted überschneiden sich bei mehreren Aufgaben, sind jedoch nicht identisch.

GNOME Disks verbindet Laufwerksinformationen, Geräteverwaltung, SMART-Anzeige, Benchmarks, Image-Funktionen und grundlegende Partitions- beziehungsweise Dateisystemverwaltung in einer grafischen Oberfläche.

GParted konzentriert sich wesentlich stärker auf die grafische Bearbeitung von Partitionen und Dateisystemstrukturen.

Innerhalb von JUMBO BLOCK Rescue ergänzen sich beide Programme daher.

Für umfangreiche Partitionsarbeiten steht GParted zur Verfügung, während GNOME Disks besonders für die schnelle Laufwerksübersicht und zusätzliche gerätebezogene Funktionen geeignet ist.

Typischer Rescue-Ablauf

Bei einem unbekannten Datenträger kann GNOME Disks zunächst zur Identifikation und Übersicht verwendet werden.

Dabei sollte zunächst ausschließlich untersucht werden:

Welcher Datenträger wurde erkannt?

Welche Kapazität besitzt er?

Welche Partitionen beziehungsweise Volumes werden angezeigt?

Welcher Laufwerkszustand wird gemeldet?

Sind SMART-Informationen verfügbar?

Erst danach sollte entschieden werden, ob weitere Diagnose-, Sicherungs- oder Reparaturschritte sinnvoll sind.

Bei einem physisch auffälligen Datenträger sollte die Priorität auf der Datensicherung liegen.

Bei einer verlorenen Partition sollte TestDisk geprüft werden.

Für umfangreiche Partitionsänderungen steht GParted zur Verfügung.

Für die spezialisierte SMART-Diagnose folgt innerhalb von JUMBO BLOCK Rescue das separate Werkzeug **SMART Diagnose**.

Grenzen

GNOME Disks ist kein spezialisiertes Datenrettungsprogramm.

Die Anwendung rekonstruiert nicht wie TestDisk verlorene Partitionierungsstrukturen und führt kein signaturbasiertes File Carving wie PhotoRec oder Foremost durch.

Die Image-Funktion ersetzt bei physisch beschädigten oder schwer lesbaren Medien nicht die spezialisierten Rettungsmechanismen von GNU ddrescue.

SMART-Werte wiederum stellen eine Diagnosehilfe dar, sind aber keine Garantie für den aktuellen oder zukünftigen fehlerfreien Zustand eines Datenträgers.

Sicherheitshinweise

GNOME Disks vereint lesende und schreibende Funktionen in derselben Oberfläche.

Deshalb muss vor jeder Operation festgestellt werden, ob sie lediglich Informationen ausliest oder den Datenträger verändert.

Besondere Vorsicht ist erforderlich bei:

Formatierung,

Partitionierung,

Änderung von Dateisystemen,

Dateisystemreparaturen,

Wiederherstellung eines Images auf einen Datenträger,

und Schreibbenchmarks.

Bei einem Datenträger mit zu rettenden Daten sollte vor solchen Operationen möglichst eine Sicherung des Ausgangszustands angelegt werden.

Bei einem physisch instabilen Medium ist GNU ddrescue für die Sicherung vorgesehen.

Quelle und Ziel müssen vor jeder schreibenden Operation eindeutig identifiziert sein.

Rolle in JUMBO BLOCK Rescue

GNOME Disks bildet innerhalb von JUMBO BLOCK Rescue ein vielseitiges **grafisches Werkzeug für Datenträger- und Blockgeräteverwaltung**.

Es ergänzt GParted um gerätebezogene Diagnose- und Verwaltungsfunktionen und stellt einen schnellen grafischen Zugang zu Laufwerksinformationen bereit.

Die funktionale Einordnung lautet:

GParted – detaillierte grafische Partitionsverwaltung

Laufwerke / GNOME Disks – Laufwerksübersicht, Geräteverwaltung, SMART-Anzeige, Benchmarks, Images und grundlegende Partitions-/Dateisystemverwaltung

SMART Diagnose – spezialisierte SMART- und Laufwerksdiagnostik

GNU ddrescue – spezialisierte Sicherung problematisch lesbarer Datenträger

TestDisk – Wiederherstellung verlorener beziehungsweise beschädigter Partitionierungsstrukturen

GNOME Disks ist damit insbesondere für die **erste technische Bestandsaufnahme eines Datenträgers** geeignet, besitzt gleichzeitig aber leistungsfähige schreibende Funktionen, die im Rescue-Betrieb nur kontrolliert eingesetzt werden dürfen.

A.8 SMART Diagnose

Bereich: Datenträger
Programm: SMART Diagnose
Technische Grundlage: smartmontools / `smartctl`
Baseline 20261001: smartmontools 7.4
Installiertes Paket: `smartmontools 7.4-3`
Bedienung: Kommandozeilenbasierte Laufwerksdiagnose; Einbindung als SMART-Diagnosefunktion in JUMBO BLOCK Rescue

Zweck

SMART Diagnose dient der technischen Zustandsanalyse unterstützter Festplatten und Solid-State-Laufwerke anhand der vom Laufwerk bereitgestellten Diagnoseinformationen.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält hierfür das Paket `smartmontools` in Version `7.4-3`.

Smartmontools stellt insbesondere die Programme `smartctl` und `smartd` bereit. `smartctl` dient der direkten Abfrage und Steuerung von SMART-Funktionen eines Laufwerks, während `smartd` für eine fortlaufende Überwachung vorgesehen ist.

Für den interaktiven Rescue-Einsatz ist vor allem **`smartctl`** relevant.

Was ist SMART?

SMART steht für:

Self-Monitoring, Analysis and Reporting Technology

Die Technik ist in vielen modernen Datenträgern integriert und stellt Informationen zur Laufwerkszuverlässigkeit und zum Gerätezustand bereit.

`smartctl` kann diese Informationen auslesen und – abhängig vom jeweiligen Gerät – interne Selbsttests des Laufwerks auslösen.

SMART ersetzt keine Datensicherung und kann nicht garantieren, dass ein Laufwerk zukünftig nicht ausfällt.

Unterstützte Geräteklassen

Smartmontools unterstützt unterschiedliche Speicherprotokolle und Geräteklassen.

Dazu gehören insbesondere:

ATA / SATA

SCSI / SAS

NVMe

Daneben existiert Unterstützung für weitere Geräte- und Controllerkonstellationen.

Welche SMART- beziehungsweise Diagnoseinformationen tatsächlich verfügbar sind, hängt vom Laufwerk, dessen Firmware, dem verwendeten Protokoll, dem Controller und gegebenenfalls einer dazwischenliegenden USB- oder RAID-Schnittstelle ab.

Gerät identifizieren

Vor jeder Diagnose muss das richtige Gerät eindeutig bestimmt werden.

Typische Linux-Gerätenamen können beispielsweise sein:

`/dev/sda`

`/dev/sdb`

oder bei NVMe:

`/dev/nvme0`

beziehungsweise

`/dev/nvme0n1`

Diese Bezeichnungen sind ausschließlich Beispiele.

Der tatsächlich zu untersuchende Datenträger muss vor dem Aufruf zweifelsfrei identifiziert werden.

Geräte automatisch suchen

`smartctl` besitzt mit

`--scan`

eine Funktion zur Suche nach unterstützten Geräten.

Beispiel:

```
sudo smartctl --scan
```

Dabei werden erkannte Geräte einschließlich des ermittelten Gerätetyps beziehungsweise Protokolls ausgegeben.

Zusätzlich steht

`--scan-open`

zur Verfügung.

Diese Variante versucht, die gefundenen Geräte auch zu öffnen, wodurch die Geräteerkennung weiter konkretisiert werden kann.

Geräteinformationen anzeigen

Mit

`-i`

beziehungsweise

`--info`

werden grundlegende Informationen über das Laufwerk ausgegeben.

Beispiel:

```
sudo smartctl -i /dev/sda
```

Bei einem ATA-Gerät können dazu unter anderem gehören:

Gerätemodell,

Seriennummer,

Firmware-Version,

Kapazität,

Informationen zum ATA-Standard,

sowie Angaben darüber, ob SMART unterstützt wird und ob SMART aktiviert ist.

Damit eignet sich dieser Aufruf zunächst zur Identifikation des Laufwerks.

SMART-Gesundheitsstatus

Mit

`-H`

beziehungsweise

`--health`

wird der SMART-Gesundheitsstatus abgefragt.

Beispiel:

```
sudo smartctl -H /dev/sda
```

Dieser Status ist eine vom Laufwerk beziehungsweise dessen SMART-Implementierung bereitgestellte Gesamtbeurteilung.

Ein positives Ergebnis bedeutet nicht, dass sämtliche Einzelwerte unauffällig sind oder ein zukünftiger Ausfall ausgeschlossen ist.

Deshalb sollte die Diagnose nicht ausschließlich auf diesen einen Status reduziert werden.

Umfangreiche SMART-Informationen

Ein zentraler Diagnoseaufruf ist:

```
sudo smartctl -a /dev/sda
```

Die Option

`-a`

beziehungsweise

`--all`

gibt eine umfangreiche Sammlung der verfügbaren SMART-Informationen aus.

Welche Informationen erscheinen, hängt vom Gerätetyp und dessen Fähigkeiten ab.

Bei einer Rescue-Diagnose ist dieser Aufruf häufig ein geeigneter Ausgangspunkt für eine detailliertere Untersuchung.

Erweiterte Gesamtausgabe

Mit

`-x`

beziehungsweise

`--xall`

kann eine noch umfassendere Ausgabe der verfügbaren SMART- und weiterer gerätebezogener Informationen angefordert werden.

Beispiel:

```
sudo smartctl -x /dev/sda
```

`-x` geht damit über die normale `-a`-Ausgabe hinaus.

Für eine technische Detailanalyse kann diese Ausgabe wertvoll sein, erzeugt jedoch entsprechend umfangreiche Diagnoseinformationen.

SMART-Attribute bei ATA-Geräten

Bei ATA/SATA-Geräten können mit

`-A`

beziehungsweise

`--attributes`

die SMART-Attribute angezeigt werden.

Beispiel:

```
sudo smartctl -A /dev/sda
```

Die ausgegebenen Attribute stammen aus der SMART-Implementierung des jeweiligen Laufwerks.

Je nach Hersteller und Modell können Bedeutung und Kodierung einzelner Rohwerte variieren.

`smartctl` verwendet unter anderem seine Laufwerksdatenbank, um bekannte Modelle und Attribute korrekt zu interpretieren.

Normalisierte Werte und Rohwerte

Bei klassischen ATA-SMART-Attributen können neben einem Rohwert auch normalisierte Werte und Schwellenwerte vorhanden sein.

Die Rohwerte dürfen nicht pauschal als physikalische Einheiten interpretiert werden.

Die Smartmontools-Dokumentation weist ausdrücklich darauf hin, dass die Umrechnung von Rohwerten in physikalische Größen nicht durch den SMART-Standard festgelegt ist und Hersteller unterschiedliche Konventionen verwenden können.

Ein einzelner Rohwert sollte deshalb immer im Kontext des Laufwerksmodells und des betreffenden Attributs beurteilt werden.

Fehlerprotokolle

SMART-fähige Geräte können Fehlerprotokolle bereitstellen.

Bei geeigneten Geräten lassen sich diese mit entsprechenden `-l`-Optionen auslesen.

Ein klassisches Beispiel ist:

```
sudo smartctl -l error /dev/sda
```

Das Fehlerprotokoll kann Informationen über vom Laufwerk registrierte Fehler enthalten.

Das Fehlen eines protokollierten Fehlers beweist jedoch nicht, dass das Medium vollständig fehlerfrei ist.

Selbsttestprotokoll

Mit

```
-l selftest
```

kann das Protokoll durchgeführter Selbsttests abgefragt werden.

Beispiel:

```
sudo smartctl -l selftest /dev/sda
```

Dort können Ergebnisse früherer interner Laufwerksselbsttests aufgeführt sein.

Nach einem neu gestarteten Selbsttest kann über dieses Protokoll später dessen Ergebnis kontrolliert werden.

Interne SMART-Selbsttests

`smartctl` kann bei unterstützten Laufwerken interne Selbsttests auslösen.

Die Tests werden vom Laufwerk selbst durchgeführt.

Welche Testarten verfügbar sind, hängt vom jeweiligen Gerät und Protokoll ab.

Für die Diagnose sind insbesondere kurze und längere beziehungsweise erweiterte Selbsttests relevant.

Kurzer Selbsttest

Bei einem unterstützten Gerät kann ein kurzer Selbsttest beispielsweise mit

```
sudo smartctl -t short /dev/sda
```

gestartet werden.

Der Test wird vom Laufwerk intern ausgeführt.

`smartctl` meldet bei unterstützten Geräten typischerweise eine erwartete Testdauer beziehungsweise den Zeitpunkt, zu dem das Ergebnis verfügbar sein sollte.

Anschließend kann das Selbsttestprotokoll erneut abgefragt werden.

Langer beziehungsweise erweiterter Selbsttest

Ein umfangreicherer Selbsttest kann bei geeigneten Geräten mit

```
sudo smartctl -t long /dev/sda
```

gestartet werden.

Dieser Test benötigt üblicherweise erheblich mehr Zeit als ein kurzer Selbsttest.

Auch hier erfolgt die eigentliche Prüfung innerhalb des Laufwerks.

Das Ergebnis kann anschließend über das Selbsttestprotokoll kontrolliert werden.

Selbsttest abbrechen

Ein laufender, nicht im Captive-Modus ausgeführter SMART-Selbsttest kann bei unterstützten Geräten mit

```
-X
```

beziehungsweise

```
--abort
```

abgebrochen werden.

Beispiel:

```
sudo smartctl -X /dev/sda
```

Ob und wie ein bestimmtes Gerät darauf reagiert, hängt von dessen Fähigkeiten ab.

Captive-Modus

Für bestimmte ATA-Selbsttests existiert der sogenannte **Captive-Modus**.

Die Smartmontools-Dokumentation warnt ausdrücklich, dass ein solcher Test das Laufwerk für die Dauer des Tests blockieren kann.

Captive-Tests sollen deshalb nur auf Laufwerken ausgeführt werden, auf denen **keine Partitionen eingehängt sind**.

Für die normale erste Rescue-Diagnose besteht in der Regel kein Grund, ohne konkrete technische Notwendigkeit einen Captive-Test zu erzwingen.

SMART aktivieren oder deaktivieren

Bei geeigneten Geräten kann SMART mit

`-s on`

aktiviert beziehungsweise mit

`-s off`

deaktiviert werden.

Beispiel:

```
sudo smartctl -s on /dev/sda
```

Dies ist bereits eine Steuerungsfunktion und keine reine Diagnoseabfrage.

Im Rescue-Betrieb sollte deshalb nicht unnötig die Gerätekonfiguration verändert werden.

Für eine erste Untersuchung ist zunächst festzustellen, ob SMART unterstützt wird und in welchem Zustand es sich befindet.

Gerätetyp vorgeben

Normalerweise versucht `smartctl`, den Gerätetyp automatisch zu erkennen.

Mit

`-d TYPE`

beziehungsweise

`--device=TYPE`

kann ein Gerätetyp ausdrücklich vorgegeben werden.

Diese Funktion ist beispielsweise bei bestimmten USB-Adaptern, RAID-Controllern oder anderen Zwischenschichten relevant.

Sie sollte jedoch nicht experimentell mit beliebigen Typangaben verwendet werden.

Für einzelne Gerätetypen warnt die Smartmontools-Dokumentation ausdrücklich vor möglichen unerwünschten Seiteneffekten bei falscher Verwendung.

USB-Gehäuse und Adapter

Bei externen SATA-Laufwerken hinter einem USB-Adapter hängt die SMART-Verfügbarkeit davon ab, ob der verwendete Bridge-Chip die erforderlichen Befehle korrekt weiterleitet und von Smartmontools unterstützt wird.

Ein Laufwerk kann daher physisch SMART unterstützen, während die Daten über einen bestimmten USB-Adapter nicht oder nicht vollständig zugänglich sind.

Ein fehlender SMART-Zugriff darf deshalb nicht automatisch als fehlende SMART-Unterstützung des eigentlichen Laufwerks interpretiert werden.

RAID-Controller

Auch RAID-Controller können den direkten Zugriff auf die darunterliegenden physischen Laufwerke verändern.

Je nach Controller sind spezielle Gerätetypen oder Zugriffsmethoden erforderlich.

Bei einem RAID-System muss daher zunächst geklärt werden, ob `smartctl` das logische Volume oder tatsächlich ein physisches Laufwerk adressiert.

Falsche Zuordnungen können zu irreführenden Diagnoseergebnissen führen.

NVMe und SMART

Smartmontools unterstützt auch NVMe-Geräte.

Die Diagnoseinformationen von NVMe entsprechen jedoch nicht einfach der klassischen ATA-SMART-Attributtabelle.

`smartctl` stellt die vom NVMe-Gerät angebotenen Gesundheits- und Fehlerinformationen entsprechend dem NVMe-Protokoll dar.

Für JUMBO BLOCK Rescue existiert zusätzlich **A.9 – NVMe Diagnose** auf Basis von `nvme-cli`.

Damit stehen für NVMe-Laufwerke zwei unterschiedliche Diagnosewege zur Verfügung:

SMART-/Health-Auswertung über Smartmontools,

sowie die native NVMe-Diagnose über `nvme-cli`.

Die native NVMe-Diagnose wird im folgenden Kapitel separat behandelt.

Temperatur

Viele Laufwerke stellen Temperaturinformationen bereit.

Diese können je nach Gerät über SMART-Attribute oder andere gerätespezifische SMART-/Health-Strukturen ausgegeben werden.

Eine erhöhte Temperatur kann diagnostisch relevant sein.

Die Interpretation muss jedoch die herstellerspezifischen Grenzwerte und die jeweilige Gerätekonstruktion berücksichtigen.

Ein einzelner Temperaturwert allein ist keine vollständige Zustandsdiagnose.

SMART ist kein Oberflächenscan

SMART Diagnose ist nicht mit einem vollständigen sektorweisen Lesetest des Datenträgers gleichzusetzen.

SMART liefert vom Laufwerk selbst bereitgestellte Gesundheits-, Fehler- und Betriebsinformationen und ermöglicht interne Selbsttests.

Eine erfolgreiche SMART-Abfrage bedeutet daher nicht automatisch, dass jeder logische Block des Datenträgers problemlos lesbar ist.

SMART ist kein Datenrettungsprogramm

`smartctl` stellt keine gelöschten Dateien wieder her.

Es rekonstruiert keine verlorenen Partitionen.

Es erzeugt auch keinen Rettungsklon eines beschädigten Datenträgers.

Seine Aufgabe ist die **Diagnose und Steuerung der vom Laufwerk bereitgestellten SMART- und verwandten Überwachungsfunktionen**.

SMART Diagnose und GNU ddrescue

SMART Diagnose und GNU ddrescue ergänzen sich.

SMART kann Hinweise darauf liefern, dass ein Laufwerk technisch auffällig ist.

GNU ddrescue kann anschließend dazu verwendet werden, möglichst viele noch lesbare Daten eines problematischen Mediums auf ein anderes Medium beziehungsweise in ein Image zu übertragen.

Bei einem bereits deutlich auffälligen Datenträger sollte nicht versucht werden, durch zahlreiche Diagnose- und Belastungstests zunächst eine vollständige Fehleranalyse zu erzwingen, bevor die wichtigen Daten gesichert werden.

Wenn der Datenerhalt Priorität besitzt, ist die Sicherung des noch lesbaren Inhalts entscheidend.

SMART Diagnose und GNOME Disks

GNOME Disks kann SMART-Daten grafisch anzeigen und SMART-Selbsttests starten.

Der Menüpunkt **SMART Diagnose** innerhalb von JUMBO BLOCK Rescue basiert dagegen auf dem spezialisierten Smartmontools-Werkzeugsatz.

Damit stehen zwei Zugriffsebenen zur Verfügung:

GNOME Disks – grafische Übersicht und grundlegende SMART-Funktionen

smartctl – detaillierte kommandozeilenbasierte SMART-Diagnose

Beide greifen auf vom Laufwerk bereitgestellte Diagnoseinformationen zurück, unterscheiden sich jedoch in Bedienung und Detailtiefe.

JSON-Ausgabe

Smartctl unterstützt strukturierte Ausgabe.

Mit

`-j`

beziehungsweise

`--json`

können Diagnoseinformationen im JSON-Format ausgegeben werden.

Beispiel:

```
sudo smartctl -a -j /dev/sda
```

Dies ist insbesondere für automatisierte Auswertungen oder die Weiterverarbeitung von Diagnoseinformationen interessant.

Für eine manuelle Rescue-Untersuchung ist die normale Textausgabe meist leichter unmittelbar zu lesen.

Laufwerksdatenbank

Smartmontools verwendet eine Laufwerksdatenbank, um bekannte Gerätemodelle und herstellerspezifische Besonderheiten zu berücksichtigen.

Bei bekannten ATA-Laufwerken können dadurch beispielsweise Modellfamilien erkannt und passende Vorgaben für die Interpretation bestimmter Attribute verwendet werden.

Diese Datenbank verbessert die Interpretation gerätespezifischer SMART-Informationen.

Grenzen der SMART-Diagnose

SMART kann wertvolle Hinweise auf den Zustand eines Datenträgers liefern, besitzt aber klare Grenzen.

Nicht jedes Laufwerk stellt dieselben Werte bereit.

Nicht jeder Hersteller interpretiert Rohwerte identisch.

USB-Adapter oder RAID-Controller können den Zugriff auf Diagnoseinformationen einschränken.

Ein Laufwerk kann trotz unauffälliger SMART-Werte Probleme besitzen oder später ausfallen.

Und ein bereits schwer beschädigtes Laufwerk sollte nicht unnötig durch umfangreiche Tests belastet werden, wenn zunächst wichtige Daten zu retten sind.

Sicherheitshinweise

Vor jeder Diagnose muss das richtige physische Laufwerk eindeutig identifiziert werden.

Reine Informationsabfragen wie `-i`, `-H`, `-A`, `-a` oder `-x` sind von Befehlen zu unterscheiden, die Funktionen des Laufwerks steuern oder Selbsttests auslösen.

Bei einem instabilen oder bereits ausfallenden Datenträger muss abgewogen werden, ob ein zusätzlicher Selbsttest sinnvoll ist oder ob zunächst eine Datenrettung durchgeführt werden sollte.

Insbesondere ein langer Selbsttest kann das Laufwerk über einen längeren Zeitraum beschäftigen.

Bei mechanisch auffälligen Laufwerken, starken Lesefehlern oder wiederholten Verbindungsabbrüchen sollte die Sicherung wichtiger Daten Priorität besitzen.

Typischer Rescue-Ablauf

Für eine erste SMART-Untersuchung bietet sich ein kontrollierter Ablauf an:

Zunächst wird der Datenträger eindeutig identifiziert.

Anschließend werden Geräteinformationen und vorhandene SMART-Daten ausgelesen.

Beispielsweise:

```
sudo smartctl -i /dev/sdX
```

danach:

```
sudo smartctl -H /dev/sdX
```

und für eine umfassendere Diagnose:

```
sudo smartctl -a /dev/sdX
```

Bei Bedarf kann zusätzlich

```
sudo smartctl -x /dev/sdX
```

verwendet werden.

Erst nach Sichtung dieser Informationen sollte entschieden werden, ob ein interner Selbsttest sinnvoll ist.

Bei Anzeichen eines ernsthaften Hardwareproblems und wichtigen zu rettenden Daten sollte stattdessen zunächst die Sicherung des Datenträgers erwogen werden.

Rolle in JUMBO BLOCK Rescue

SMART Diagnose bildet innerhalb von JUMBO BLOCK Rescue die spezialisierte Ebene zur **Zustandsanalyse SMART-fähiger Speichergeräte**.

Die Werkzeuge des Bereichs Datenträger ergänzen sich damit funktional:

GParted – Partitionen und Partitionstabellen grafisch verwalten

Laufwerke / GNOME Disks – grafische Laufwerksübersicht, Geräteverwaltung, SMART-Anzeige, Benchmarks und Images

SMART Diagnose / smartctl – detaillierte SMART- und Laufwerksdiagnostik

NVMe Diagnose / nvme-cli – native Diagnose und Verwaltung von NVMe-Geräten

Im Rescue-Ablauf kann SMART Diagnose damit insbesondere bei der Entscheidung helfen, ob ein Datenträger technisch unauffällig erscheint oder Hinweise auf einen problematischen Zustand liefert.

Die SMART-Auswertung ist jedoch immer eine **Diagnosehilfe und keine Ausfallgarantie**.

Bei gefährdeten Daten bleibt die möglichst frühzeitige Sicherung des Mediums der entscheidende Schritt.

A.9 NVMe Diagnose

Bereich: Datenträger
Programm: NVMe Diagnose
Technische Grundlage: `nvme-cli`
Baseline 20261001: nvme-cli 2.13
Installiertes Paket: `nvme-cli 2.13-2`
Bedienung: Kommandozeile

Zweck

NVMe Diagnose dient der direkten Untersuchung von **NVMe-Speichergeräten** über die vom NVMe-Protokoll bereitgestellten Verwaltungs- und Diagnosefunktionen.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält hierfür:

`nvme-cli 2.13-2`

Das zentrale Kommando lautet:

`nvme`

Mit `nvme-cli` können NVMe-Controller und Namespaces identifiziert, Geräteinformationen ausgelesen, SMART-/Health-Informationen abgefragt, Fehlerprotokolle untersucht und – sofern vom Gerät unterstützt – interne Selbsttests ausgeführt werden.

Das Werkzeug besitzt darüber hinaus zahlreiche administrative Funktionen, die NVMe-Geräte verändern können.

Im Rescue-Betrieb ist deshalb strikt zwischen **reiner Diagnose** und **verändernden NVMe-Kommandos** zu unterscheiden.

NVMe

NVMe steht für:

Non-Volatile Memory Express

NVMe ist eine Schnittstellen- und Befehlsspezifikation für nichtflüchtige Speichergeräte.

Unter Linux erscheinen NVMe-Geräte typischerweise mit Gerätenamen wie:

`/dev/nvme0`

und zugehörige Namespaces beispielsweise als:

`/dev/nvme0n1`

Diese Namen sind lediglich Beispiele.

Vor jeder Diagnose muss das tatsächlich zu untersuchende Gerät eindeutig identifiziert werden.

Controller und Namespace

Bei NVMe muss zwischen **Controller** und **Namespace** unterschieden werden.

Ein Controller kann einen oder mehrere Namespaces bereitstellen.

Ein Gerätename wie

`/dev/nvme0`

bezeichnet typischerweise den NVMe-Controller.

Ein Gerätename wie

`/dev/nvme0n1`

bezeichnet einen Namespace dieses Controllers.

Diese Unterscheidung ist für die korrekte Verwendung von `nvme-cli` wesentlich.

Einige Befehle akzeptieren sowohl ein NVMe-Character-Device wie `/dev/nvme0` als auch ein Namespace-Blockgerät wie `/dev/nvme0n1`. Welche Form sinnvoll beziehungsweise erforderlich ist, hängt vom jeweiligen Befehl ab.

Erkannte NVMe-Geräte auflisten

Ein zentraler erster Befehl ist:

```
sudo nvme list
```

Er dient dazu, vom System erkannte NVMe-Geräte aufzulisten.

Damit lässt sich zunächst feststellen, welche NVMe-Speichergeräte beziehungsweise Namespaces vorhanden sind.

Vor allen weiteren Operationen sollte anhand dieser Informationen sichergestellt werden, dass das richtige Gerät ausgewählt wurde.

Controller identifizieren

Für detaillierte Informationen über einen NVMe-Controller steht in nvme-cli 2.13 der Befehl

```
nvme id-ctrl
```

zur Verfügung.

Beispiel:

```
sudo nvme id-ctrl /dev/nvme0
```

Der Befehl sendet ein **Identify Controller**-Kommando an das NVMe-Gerät und gibt die zurückgelieferte Controller-Datenstruktur aus.

Diese enthält zahlreiche technische Eigenschaften des Controllers.

Dazu können – abhängig vom Gerät – beispielsweise Identifikations-, Firmware- und Fähigkeitsinformationen gehören.

Für eine Rescue-Untersuchung ist dies eine wichtige Quelle für die technische Identifikation des NVMe-Geräts.

Namespace identifizieren

Die Informationen eines NVMe-Namespace können mit

`nvme id-ns`

abgerufen werden.

Beispiel:

```
sudo nvme id-ns /dev/nvme0n1
```

Damit wird die vom NVMe-Gerät bereitgestellte **Identify Namespace**-Datenstruktur ausgegeben.

Controller- und Namespace-Informationen erfüllen unterschiedliche Aufgaben und sollten deshalb nicht miteinander verwechselt werden.

SMART-/Health-Informationen

NVMe stellt eine eigene **SMART / Health Information Log Page** bereit.

In nvme-cli 2.13 wird diese mit

`nvme smart-log`

abgerufen.

Beispiel:

```
sudo nvme smart-log /dev/nvme0
```

Der Befehl fordert die NVMe-SMART-Log-Page an und gibt die zurückgelieferte Struktur aus.

In Version 2.13 kann sowohl ein NVMe-Controllergerät wie

`/dev/nvme0`

als auch ein Namespace-Blockgerät wie

`/dev/nvme0n1`

angegeben werden.

Globale und Namespace-bezogene SMART-Daten

`nvme smart-log` unterstützt die Option

`--namespace-id`

beziehungsweise

`-n`.

Damit kann eine bestimmte Namespace-ID angefordert werden.

Ohne eine solche Angabe verwendet nvme-cli 2.13 für die globale SMART-Abfrage standardmäßig die Namespace-ID `0xffffffff`.

Ob ein Gerät SMART-/Health-Informationen tatsächlich namespacebezogen bereitstellt, hängt von den Fähigkeiten des jeweiligen Controllers ab.

Typische Health-Informationen

Die SMART-/Health-Log-Page eines NVMe-Geräts enthält standardisierte Gesundheits- und Betriebsinformationen.

Abhängig von Gerät und Implementierung können dabei unter anderem Informationen zu folgenden Bereichen ausgegeben werden:

kritische Warnungen,

Temperatur,

verfügbare Reservekapazität,

verbrauchte Lebensdauer,

gelesene und geschriebene Datenmengen,

Betriebsstunden,

Ein-/Ausschaltvorgänge,

unsichere Abschaltungen,

Medien- und Datenintegritätsfehler,

sowie Fehlerprotokolleinträge.

Diese Werte werden vom NVMe-Gerät beziehungsweise Controller bereitgestellt.

Sie sind Diagnoseinformationen und keine Garantie dafür, dass das Laufwerk fehlerfrei ist oder nicht kurzfristig ausfallen kann.

Critical Warning

Besondere Aufmerksamkeit verdient das Feld **Critical Warning** der NVMe-SMART-/Health-Informationen.

Es dient dazu, bestimmte vom Controller erkannte kritische Zustände zu signalisieren.

Ein gesetztes Warnbit sollte nicht isoliert betrachtet werden, sondern zusammen mit den übrigen Health-Informationen und der technischen Situation des Datenträgers ausgewertet werden.

Bei einem Laufwerk mit wichtigen Daten und Hinweisen auf einen kritischen Zustand sollte die Datensicherung Priorität erhalten.

Temperatur

NVMe-Controller können Temperaturinformationen über die SMART-/Health-Log-Page bereitstellen.

Eine ungewöhnliche oder kritische Temperatur kann diagnostisch relevant sein.

Die Temperatur allein erlaubt jedoch keine vollständige Beurteilung des Laufwerkszustands.

Sie muss zusammen mit den übrigen Health-Daten und den vom Gerät gemeldeten Warnzuständen betrachtet werden.

Available Spare

NVMe kann Informationen über die noch verfügbare Reservekapazität des nichtflüchtigen Speichers bereitstellen.

Dazu gehören unter anderem Werte für die verfügbare Reserve und den zugehörigen Schwellenwert.

Ein kritischer Zustand der Reserve kann über die Health-Informationen signalisiert werden.

Percentage Used

Das NVMe-Health-Log enthält das Feld **Percentage Used**.

Dieses beschreibt eine vom Controller bereitgestellte Schätzung der bereits verbrauchten Lebensdauer beziehungsweise Endurance des NVM-Subsystems.

Dieser Wert ist nicht mit dem prozentual belegten Speicherplatz eines Dateisystems zu verwechseln.

Er betrifft die vom Controller bewertete Nutzung beziehungsweise Abnutzung des Speichermediums.

Data Units Read und Data Units Written

NVMe kann Zähler für die vom Host gelesenen und geschriebenen Datenmengen bereitstellen.

Diese Werte ermöglichen eine Beurteilung der bisherigen Nutzung des Laufwerks.

Sie sind Betriebszähler und nicht unmittelbar ein Fehlerindikator.

Power Cycles und Power On Hours

NVMe-Geräte können unter anderem die Anzahl der Einschaltzyklen und Betriebsstunden melden.

Diese Werte liefern zusätzliche Informationen über die bisherige Nutzung des Geräts.

Auch hier gilt:

Ein hoher Betriebsstundenzähler allein beweist keinen Defekt.

Er muss im Zusammenhang mit den übrigen Health- und Fehlerdaten interpretiert werden.

Unsafe Shutdowns

Die SMART-/Health-Informationen können einen Zähler für **Unsafe Shutdowns** enthalten.

Dieser dokumentiert entsprechende vom NVMe-Subsystem erfasste Abschaltvorgänge.

Der Wert kann bei der Untersuchung eines Systems Hinweise auf dessen Betriebshistorie liefern.

Er stellt für sich allein jedoch noch keine vollständige Fehlerdiagnose dar.

Media and Data Integrity Errors

Besonders relevant für einen Rescue-Fall sind vom NVMe-Gerät gemeldete **Media and Data Integrity Errors**.

Ein entsprechender Zähler weist darauf hin, dass das Gerät solche Fehler registriert hat.

Bei gleichzeitig wichtigen Daten sollte dies als ernstzunehmender Diagnosehinweis behandelt und die weitere Belastung des Laufwerks gegen die Notwendigkeit einer möglichst schnellen Datensicherung abgewogen werden.

Fehlerprotokoll

NVMe besitzt ein eigenes Fehlerprotokoll.

In nvme-cli 2.13 kann dieses mit

```
nvme error-log
```

abgerufen werden.

Beispiel:

```
sudo nvme error-log /dev/nvme0
```

Der Befehl fordert die NVMe **Error Information Log Page** an und gibt die entsprechenden Einträge aus.

Das Fehlerprotokoll kann zusätzliche Informationen zu vom Controller registrierten NVMe-Fehlern enthalten.

Anzahl der Fehlerprotokolleinträge

`nvme error-log` unterstützt in Version 2.13 die Angabe der Anzahl abzurufender Einträge.

Dafür steht

```
--log-entries
```

beziehungsweise

```
-e
```

zur Verfügung.

Beispiel:

```
sudo nvme error-log /dev/nvme0 -e 16
```

Damit können gezielt mehrere Einträge des NVMe-Fehlerprotokolls angefordert werden.

SMART-Log im JSON-Format

nvme-cli 2.13 unterstützt für `smart-log` unterschiedliche Ausgabeformate.

Mit

```
-o json
```

beziehungsweise

```
--output-format=json
```

kann eine strukturierte JSON-Ausgabe angefordert werden.

Beispiel:

```
sudo nvme smart-log /dev/nvme0 -o json
```

Dies ist insbesondere für automatisierte Auswertungen und die maschinelle Weiterverarbeitung von Diagnoseinformationen geeignet.

Daneben unterstützt `smart-log` in Version 2.13 normale und binäre Ausgabe.

Rohdaten ausgeben

Mit

```
--raw-binary
```

beziehungsweise

```
-b
```

kann `nvme smart-log` die rohe SMART-Log-Datenstruktur ausgeben.

Die offizielle Dokumentation empfiehlt bei dieser Betriebsart sinngemäß, die Binärausgabe in eine Datei beziehungsweise an einen geeigneten Empfänger umzuleiten.

Für die normale manuelle Diagnose ist die menschenlesbare Ausgabe wesentlich geeigneter.

Device Self-Test

NVMe definiert einen **Device Self-Test**.

nvme-cli 2.13 stellt dafür den Befehl

```
nvme device-self-test
```

bereit.

Beispiel:

```
sudo nvme device-self-test /dev/nvme0 -s 1
```

Der Befehl löst einen vom NVMe-Gerät selbst durchgeführten Test aus.

Ob diese Funktion verfügbar ist, hängt von den Fähigkeiten des jeweiligen Geräts ab.

Kurzer Selbsttest

In nvme-cli 2.13 bedeutet der Self-Test-Code

`1`

den Start eines **Short Device Self-Test**.

Beispiel:

```
sudo nvme device-self-test /dev/nvme0 -s 1
```

Bei Bedarf kann mit `-n` beziehungsweise `--namespace-id` ein Namespace angegeben werden.

Erweiterter Selbsttest

Der Self-Test-Code

`2`

startet einen **Extended Device Self-Test**.

Beispiel:

```
sudo nvme device-self-test /dev/nvme0 -s 2
```

Ein erweiterter Selbsttest kann wesentlich länger dauern und das Gerät entsprechend länger beschäftigen.

Bei einem bereits instabilen Laufwerk mit wichtigen zu rettenden Daten sollte deshalb zunächst geprüft werden, ob die Durchführung eines solchen Tests tatsächlich sinnvoll ist oder ob die Datensicherung Vorrang haben muss.

Status des Selbsttests

Der Self-Test-Code

`0`

fragt den aktuellen Zustand einer Device-Self-Test-Operation ab.

Damit lässt sich feststellen, ob beziehungsweise in welchem Zustand sich ein entsprechender Selbsttest befindet.

Selbsttest abbrechen

Der Self-Test-Code

`0xf`

beziehungsweise `fh` dient zum Abbruch einer laufenden Device-Self-Test-Operation.

Beispiel:

```
sudo nvme device-self-test /dev/nvme0 -s 0xf
```

Auf Selbsttest warten

nvme-cli 2.13 unterstützt beim Device Self-Test zusätzlich

`--wait`

beziehungsweise

`-w`.

Damit wartet das Programm auf den Abschluss des Device Self-Test, bevor es beendet wird.

Ohne diese Option kann der Test gestartet werden, während die spätere Ergebnisabfrage getrennt erfolgt.

Selbsttestprotokoll

Die Ergebnisse von Device-Self-Tests können in nvme-cli 2.13 über

`nvme self-test-log`

abgerufen werden.

Beispiel:

```
sudo nvme self-test-log /dev/nvme0
```

Damit können die vom Gerät bereitgestellten Informationen zu durchgeführten Selbsttests untersucht werden.

Abgrenzung zur SMART Diagnose

A.8 und A.9 überschneiden sich teilweise, weil Smartmontools ebenfalls NVMe-Geräte unterstützt.

Die Werkzeuge arbeiten jedoch auf unterschiedlichen Ebenen.

SMART Diagnose / smartctl

stellt eine einheitliche Diagnoseoberfläche für verschiedene Laufwerkstypen wie ATA, SCSI und NVMe bereit.

NVMe Diagnose / nvme-cli

greift unmittelbar auf die NVMe-spezifischen Verwaltungs-, Identify-, Log- und Diagnosefunktionen zu.

Für eine gezielte NVMe-Untersuchung bietet `nvme-cli` deshalb Informationen und Befehle, die über eine allgemeine SMART-Abfrage hinausgehen.

Diagnose versus Administration

`nvme-cli` ist nicht ausschließlich ein Diagnoseprogramm.

Das Werkzeug enthält auch administrative und potenziell destruktive Befehle.

Dazu gehören beispielsweise Funktionen für:

Formatierung,

Namespace-Erstellung und -Löschung,

Namespace-Zuordnung,

Firmware-Download und -Aktivierung,

Schreiboperationen,

Write Zeroes,

Data Set Management,

Feature-Änderungen,

Controller-Reset,

sowie weitere NVMe-Administrationsbefehle.

Diese Funktionen gehören zum Leistungsumfang von nvme-cli, sind aber **nicht Bestandteil einer unverbindlichen Rescue-Diagnose**.

Format

Besondere Vorsicht ist beim Befehl

`nvme format`

erforderlich.

Dieser führt eine NVMe-Formatierungsoperation aus.

Je nach verwendeten Optionen und Gerätefähigkeiten kann eine solche Operation die Nutzdaten eines Namespace verändern beziehungsweise löschen.

`nvme format` darf daher niemals als Diagnosebefehl behandelt werden.

Auf einem Originaldatenträger mit zu rettenden Daten ist eine solche Operation ohne eindeutige technische Absicht nicht zulässig.

Namespace-Verwaltung

nvme-cli besitzt unter anderem Befehle zum Erstellen, Löschen, Anhängen und Trennen von Namespaces.

Diese Funktionen verändern die logische NVMe-Konfiguration.

Sie sind administrative Werkzeuge und für eine reine Zustandsdiagnose nicht erforderlich.

Im Rescue-Betrieb dürfen sie nicht experimentell auf einem Datenträger mit wichtigen Daten eingesetzt werden.

Schreibbefehle

nvme-cli enthält direkte I/O-Kommandos wie

`nvme write`

und

`nvme write-zeroes`.

Solche Befehle schreiben auf das NVMe-Medium.

Sie sind damit grundsätzlich von rein lesenden Diagnosebefehlen zu unterscheiden.

Bei einem Originalmedium mit zu rettenden Daten dürfen solche Befehle nicht ohne ausdrückliche technische Notwendigkeit ausgeführt werden.

Firmware-Funktionen

nvme-cli unterstützt auch NVMe-Firmwareoperationen.

Dazu gehören Funktionen zum Herunterladen und Aktivieren beziehungsweise Committen von Firmware.

Firmwareoperationen verändern den Betriebszustand eines Geräts und können zusätzliche Risiken verursachen.

Sie gehören nicht zu einer normalen Rescue-Erstdiagnose.

Controller Reset

Auch ein NVMe-Controller-Reset ist mit nvme-cli möglich.

Ein Reset verändert zwar nicht mit derselben Zielsetzung wie eine Formatierung die gespeicherten Nutzdaten, greift aber aktiv in den Betriebszustand des Controllers ein.

Ein solcher Befehl sollte deshalb nicht mit einer reinen Informationsabfrage gleichgesetzt werden.

Sichere erste Diagnose

Für eine erste Untersuchung eines NVMe-Geräts sind insbesondere lesende beziehungsweise diagnostische Befehle geeignet.

Ein typischer Ablauf ist:

```
sudo nvme list
```

zur Identifikation der vorhandenen NVMe-Geräte,

anschließend beispielsweise

```
sudo nvme id-ctrl /dev/nvme0
```

zur Identifikation des Controllers,

danach

```
sudo nvme smart-log /dev/nvme0
```

für die Health-Informationen,

und

```
sudo nvme error-log /dev/nvme0
```

für das Fehlerprotokoll.

Erst nach Auswertung dieser Informationen sollte entschieden werden, ob ein Device Self-Test sinnvoll ist.

Verhalten bei auffälligen NVMe-Laufwerken

Zeigt ein NVMe-Gerät kritische Warnungen, Medienfehler, wiederholte I/O-Fehler, Verbindungsprobleme oder andere deutliche Auffälligkeiten, muss der Zweck der weiteren Arbeit berücksichtigt werden.

Wenn wichtige Daten gerettet werden sollen, ist eine umfangreiche Diagnose nicht zwangsläufig die höchste Priorität.

Jeder zusätzliche Test beansprucht das Gerät.

Bei einem instabilen Medium kann es sinnvoller sein, zunächst die noch lesbaren Daten zu sichern.

NVMe Diagnose und GNU ddrescue

GNU ddrescue und nvme-cli erfüllen unterschiedliche Aufgaben.

nvme-cli untersucht NVMe-spezifische Geräteinformationen, Health-Daten, Fehlerprotokolle und weitere Controllerfunktionen.

GNU ddrescue versucht, lesbare Daten eines problematischen Blockgeräts auf ein anderes Medium beziehungsweise in ein Image zu übertragen.

nvme-cli diagnostiziert damit den Zustand.

ddrescue dient der eigentlichen blockweisen Datensicherung beziehungsweise Datenrettung.

Grenzen

NVMe Diagnose kann keine gelöschten Dateien rekonstruieren.

Sie stellt keine verlorenen Partitionen wieder her.

Sie ersetzt keinen Rettungsklon.

Und auch ein unauffälliges SMART-/Health-Log garantiert nicht, dass sämtliche Daten des Laufwerks lesbar sind oder das Gerät zukünftig nicht ausfällt.

Umgekehrt muss ein einzelner hoher Betriebszähler nicht automatisch einen Defekt bedeuten.

Die Diagnoseinformationen müssen im Zusammenhang bewertet werden.

Sicherheitshinweise

Vor jedem `nvme`-Befehl muss das Zielgerät eindeutig identifiziert sein.

Besonders wichtig ist die Unterscheidung zwischen:

`/dev/nvmeX`

für einen Controller

und

`/dev/nvmeXnY`

für einen Namespace.

Bei Rescue-Arbeiten sollte außerdem vor der Ausführung jedes unbekannten Befehls geprüft werden, ob dieser lediglich Informationen liest oder den Controller beziehungsweise das Medium verändert.

Besonders kritisch sind Formatierungs-, Namespace-, Firmware-, Schreib- und sonstige administrative Befehle.

Bei einem beschädigten Originalmedium gilt:

Diagnose darf die Rettung der noch verfügbaren Daten nicht unnötig gefährden.

Versionshinweis

Dieses Kapitel dokumentiert ausdrücklich **nvme-cli 2.13**, entsprechend der JUMBO BLOCK Rescue Baseline 20261001.

Die in diesem Kapitel verwendeten Befehlsformen wie

`nvme id-ctrl`

`nvme smart-log`

`nvme error-log`

und

`nvme self-test-log`

entsprechen der Dokumentation von nvme-cli 2.13.

Spätere nvme-cli-Versionen haben Teile der Befehlsstruktur verändert.

Diese späteren Änderungen werden bewusst **nicht rückwirkend auf die Baseline 20261001 übertragen**.

Rolle in JUMBO BLOCK Rescue

NVMe Diagnose bildet innerhalb von JUMBO BLOCK Rescue die spezialisierte Diagnoseebene für **NVMe-Speichergeräte**.

Die vier Werkzeuge des Bereichs Datenträger ergänzen sich damit:

GParted – grafische Partitions- und Dateisystemverwaltung

Laufwerke / GNOME Disks – grafische Geräteübersicht, Datenträgerverwaltung, Images, Benchmarks und grundlegende Diagnose

SMART Diagnose / smartctl – protokollübergreifende Laufwerksdiagnostik für unter anderem ATA, SCSI und NVMe

NVMe Diagnose / nvme-cli – native NVMe-Controller-, Namespace-, Health-, Fehler- und Selbsttestdiagnostik

Für NVMe-Systeme liefert `nvme-cli` damit die detaillierteste direkte Schnittstelle zu den vom NVMe-Protokoll bereitgestellten Diagnoseinformationen.

Gleichzeitig ist es ein leistungsfähiges Administrationswerkzeug.

Gerade deshalb ist im Rescue-Einsatz die konsequente Trennung zwischen **lesender Diagnose und verändernden Kommandos** zwingend.

A.10 ClamAV Scan

Bereich: Sicherheit
Programm: ClamAV Scan
Technische Grundlage: ClamAV / `clamscan`
Baseline 20261001: ClamAV 1.4.3
Installierte Kernpakete: `clamav 1.4.3+dfsg-1`, `clamav-base 1.4.3+dfsg-1`, `libclamav12 1.4.3+dfsg-1`
Bedienung: Kommandozeilenbasierter Malware-Scanner

Zweck

ClamAV Scan dient innerhalb von JUMBO BLOCK Rescue zur Untersuchung von Dateien und Verzeichnissen auf Schadsoftware und andere durch die geladenen ClamAV-Signaturen beziehungsweise Erkennungsmechanismen identifizierbare Inhalte.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält ClamAV in Version 1.4.3.

Für den unmittelbaren, einmaligen Scan ist insbesondere das Kommando

`clamscan`

relevant.

`clamscan` ist ein eigenständiger Kommandozeilenscanner. Er benötigt keinen laufenden `clamd`-Daemon. Bei jedem Aufruf wird eine Scan-Engine erzeugt, die vorhandene Virendatenbank geladen, das angegebene Ziel untersucht, ein Scanbericht ausgegeben und der Prozess anschließend beendet.

Diese Arbeitsweise eignet sich besonders für ein Rescue-System, in dem Datenträger, Verzeichnisse oder einzelne Dateien gezielt untersucht werden sollen.

Signaturdatenbank

ClamAV benötigt eine vorhandene Signaturdatenbank, damit die Scan-Engine arbeiten kann.

`clamscan` sucht standardmäßig an dem Ort nach den Datenbanken, an dem die ClamAV-Signaturen durch `freshclam` installiert beziehungsweise aktualisiert wurden.

Damit müssen zwei Funktionen unterschieden werden:

ClamAV Scan / `clamscan` untersucht Dateien.

ClamAV Signaturen / `freshclam` aktualisiert die offiziellen ClamAV-Signaturdatenbanken.

Die Signaturverwaltung wird deshalb separat in **A.11 – ClamAV Signaturen** dokumentiert.

Grundsyntax

Die grundlegende Syntax lautet:

```
clamscan [Optionen] [Datei/Verzeichnis/-]
```

Damit können einzelne Dateien, Verzeichnisse oder andere geeignete Eingaben untersucht werden.

Einzelne Datei scannen

Eine einzelne Datei kann direkt angegeben werden.

Beispiel:

```
clamscan /pfad/zur/datei
```

ClamAV untersucht die angegebene Datei und gibt anschließend das Ergebnis aus.

Dies ist insbesondere geeignet, wenn eine verdächtige Datei gezielt überprüft werden soll.

Verzeichnis scannen

Ein Verzeichnis kann ebenfalls direkt angegeben werden.

Beispiel:

```
clamscan /pfad/zum/verzeichnis
```

Ohne zusätzliche Rekursionsoption untersucht `clamscan` dabei nicht automatisch sämtliche Unterverzeichnisse.

Für einen vollständigen Verzeichnisbaum muss die rekursive Suche ausdrücklich aktiviert werden.

Rekursiver Scan

Mit

```
--recursive
```

beziehungsweise

```
-r
```

werden Unterverzeichnisse rekursiv durchsucht.

Beispiel:

```
clamscan -r /pfad/zum/verzeichnis
```

oder:

```
clamscan --recursive /pfad/zum/verzeichnis
```

Für die Untersuchung eines eingehängten fremden Datenträgers ist dies eine zentrale Funktion.

Ist ein zu untersuchendes Dateisystem beispielsweise unter einem geeigneten Mountpunkt eingebunden, kann dessen Verzeichnisbaum gezielt rekursiv gescannt werden.

Gesamtes System scannen

Technisch kann `clamscan` auch auf das Wurzelverzeichnis angesetzt werden.

Beispiel:

```
clamscan --recursive /
```

Damit wird ein sehr umfangreicher Scan ausgelöst.

ClamAV weist ausdrücklich darauf hin, dass ein vollständiger Systemscan lange dauern kann.

ClamAV besitzt keinen speziellen „Quick Scan“-Modus. Der Anwender bestimmt selbst, welche Dateien und Verzeichnisse untersucht werden.

Für JUMBO BLOCK Rescue ist deshalb ein gezielter Scan des tatsächlich relevanten Datenträgers oder Verzeichnisses häufig zweckmäßiger als ein unselektierter Scan des gesamten laufenden Rescue-Systems.

Scan eines eingehängten Datenträgers

Ein typisches Rescue-Szenario besteht darin, ein fremdes Dateisystem einzuhängen und anschließend dessen Mountpunkt zu untersuchen.

Die genaue Pfadangabe hängt vom tatsächlich verwendeten Mountpunkt ab.

Das Grundprinzip lautet:

```
clamscan -r /pfad/zum/mountpunkt
```

Vor dem Scan muss sichergestellt sein, dass tatsächlich der gewünschte Datenträger untersucht wird.

Die Bezeichnung des Mountpunkts darf nicht geraten werden.

Fundmeldungen

Erkennt ClamAV eine Datei anhand der geladenen Erkennungsdaten als schädlich oder verdächtig, wird dies im Scanergebnis gemeldet.

Eine solche Meldung muss anschließend bewertet werden.

ClamAV selbst weist darauf hin, dass **False Positives** möglich sind.

Ein Scan-Fund bedeutet deshalb nicht, dass eine Datei ohne weitere Prüfung sofort gelöscht werden sollte.

Scan-Zusammenfassung

Nach Abschluss eines `clamscan`-Laufs wird eine Zusammenfassung ausgegeben.

Diese dient dazu, den Verlauf und das Ergebnis des Scans zu beurteilen.

Bei umfangreichen Rescue-Untersuchungen kann es sinnvoll sein, zusätzlich ein Scanprotokoll in einer Datei zu speichern.

Scanbericht in Datei schreiben

Mit

`--log=FILE`

kann der Scanbericht in einer Datei gespeichert werden.

Beispiel:

```
clamscan -r --log=/pfad/scan.log /pfad/zum/scan
```

Dies ist bei Rescue- und Analysearbeiten besonders nützlich, weil das Ergebnis anschließend dokumentiert und getrennt ausgewertet werden kann.

Der Speicherort des Protokolls sollte bewusst gewählt werden.

Nur infizierte Dateien anzeigen

`clamscan` unterstützt die Option

`--infected`

beziehungsweise

`-i`.

Damit wird die normale Ausgabe reduziert, sodass nur infizierte Dateien ausgegeben werden.

Beispiel:

```
clamscan -r -i /pfad/zum/scan
```

Dies kann bei sehr großen Datenbeständen die Übersicht verbessern.

Quiet-Modus

Mit

`--quiet`

wird die Ausgabe weiter reduziert.

ClamAV dokumentiert diese Option als Ausgabe lediglich von Fehlermeldungen.

Für eine interaktive Analyse ist dieser Modus nicht immer sinnvoll, kann jedoch bei automatisierten Abläufen oder gezielter Protokollierung nützlich sein.

Akustische Fundmeldung

Mit

`--bell`

kann ClamAV bei einer Viruserkennung ein akustisches Signal ausgeben.

Diese Funktion verändert den Scan selbst nicht, sondern dient lediglich der Benachrichtigung während eines interaktiven Scanvorgangs.

Alternative Signaturdatenbank

Mit

`--database=FILE/DIR`

kann ausdrücklich eine bestimmte Signaturdatei oder ein Verzeichnis mit unterstützten ClamAV-Datenbanken geladen werden.

Damit kann `clamscan` bei Bedarf mit einer gezielt ausgewählten Signaturbasis betrieben werden.

Für den normalen JUMBO BLOCK Rescue Einsatz ist die standardmäßig installierte beziehungsweise mit `freshclam` gepflegte Datenbank der vorgesehene Ausgangspunkt.

Nur offizielle Signaturen

`clamscan` unterstützt

`--official-db-only`

Damit kann festgelegt werden, dass ausschließlich offizielle ClamAV-Signaturen geladen werden.

Dies ist insbesondere dann relevant, wenn neben den offiziellen Datenbanken zusätzliche Signaturdateien vorhanden wären.

PUA-Erkennung

ClamAV unterstützt die Erkennung sogenannter **Potentially Unwanted Applications (PUA)**.

Für `clamscan` steht dafür

`--detect-pua`

zur Verfügung.

Zusätzlich können PUA-Kategorien mit

`--include-pua=KATEGORIE`

beziehungsweise

`--exclude-pua=KATEGORIE`

ein- oder ausgeschlossen werden.

PUA ist von einer eindeutigen Malware-Klassifikation zu unterscheiden. Der Begriff umfasst Software, die aufgrund bestimmter Eigenschaften oder Verhaltensweisen als potenziell unerwünscht eingestuft wird.

Eine entsprechende Meldung muss deshalb im jeweiligen Kontext bewertet werden.

Archive und Container

Die ClamAV-Scan-Engine kann unterstützte Container- und Archivformate analysieren.

Bei solchen Dateien muss berücksichtigt werden, dass die tatsächlich zu untersuchende Datenmenge erheblich größer sein kann als die äußere Archivdatei.

ClamAV besitzt deshalb Größen- und Scanlimits, mit denen die Verarbeitung solcher Inhalte begrenzt wird.

Maximale Dateigröße

Mit

`--max-filesize=GRÖSSE`

kann die maximale Größe einer Datei festgelegt werden, die ClamAV scannt.

Dateien oberhalb dieser Grenze werden übersprungen und von `clamscan` als sauber angenommen.

Dies ist für die Interpretation eines Scanergebnisses wichtig:

„Nicht als infiziert gemeldet“ ist nicht automatisch gleichbedeutend mit „vollständig untersucht“.

Scanlimits müssen bei einer sicherheitsrelevanten Bewertung berücksichtigt werden.

Maximale Scanmenge pro Container

Mit

`--max-scansize=GRÖSSE`

wird begrenzt, wie viele Daten innerhalb einer Containerdatei maximal untersucht werden.

Dies betrifft beispielsweise die Verarbeitung komplexer beziehungsweise großer Containerstrukturen.

Auch hier kann ein gesetztes Limit dazu führen, dass nicht sämtliche theoretisch enthaltenen Daten untersucht werden.

Dateiliste verwenden

Mit

`--file-list=DATEI`

kann eine Datei angegeben werden, die die zu untersuchenden Pfade enthält.

Dies ermöglicht gezielte Scanlisten, ohne jeden Pfad einzeln als Kommandozeilenargument anzugeben.

Dateisystemgrenzen

Für rekursive Scans unterstützt `clamscan`

`--cross-fs=yes`

beziehungsweise

`--cross-fs=no`.

Damit lässt sich steuern, ob beim rekursiven Durchlaufen eines Verzeichnisbaums auf andere Dateisysteme gewechselt wird.

Standardmäßig ist `--cross-fs=yes` aktiviert.

Für ein Rescue-System ist diese Option besonders relevant.

Wird ein Verzeichnisbaum untersucht, unterhalb dessen weitere Dateisysteme eingehängt sind, kann der Scan andernfalls über den ursprünglich vorgesehenen Datenträger hinausreichen.

Bei gezielten Untersuchungen sollte deshalb bewusst entschieden werden, ob Dateisystemgrenzen überschritten werden sollen.

Infizierte Dateien verschieben

Mit

`--move=VERZEICHNIS`

kann ClamAV Dateien, die einen Fund auslösen, in ein angegebenes Verzeichnis verschieben.

Beispielprinzip:

```
clamscan -r --move=/pfad/quarantaene /pfad/zum/scan
```

Diese Funktion verändert den untersuchten Datenbestand.

Sie ist deshalb nicht mit einer rein lesenden Diagnose gleichzusetzen.

Bei einem fremden Originaldatenträger oder einem Beweissicherungs-/Rescue-Szenario sollte eine solche Veränderung nur bewusst vorgenommen werden.

Infizierte Dateien kopieren

Mit

`--copy=VERZEICHNIS`

können Dateien mit Fundmeldung in ein angegebenes Verzeichnis kopiert werden.

Im Gegensatz zu `--move` bleibt die ursprüngliche Datei dabei am Ausgangsort bestehen.

Auch hier muss berücksichtigt werden, dass mit potenziell schädlichen Dateien gearbeitet wird.

Das Zielverzeichnis sollte entsprechend kontrolliert behandelt werden.

Dateien automatisch löschen

ClamAV besitzt eine Option

`--remove`

zum Löschen von Dateien, die während eines Scans einen Alarm auslösen.

Diese Option ist für einen Rescue-Einsatz besonders kritisch.

Die offizielle ClamAV-Dokumentation warnt ausdrücklich davor, diese Funktion allgemein einzusetzen, weil **False Positives auftreten können**.

ClamAV empfiehlt stattdessen eher, Dateien zu verschieben oder zu kopieren und einen Fund anschließend zu prüfen.

Für JUMBO BLOCK Rescue gilt daher:

Ein automatisches Löschen erkannter Dateien ist kein geeigneter Standardablauf.

Die Diagnose sollte zunächst erhalten bleiben und der Fund bewertet werden.

Heuristische Warnungen

ClamAV unterstützt heuristische Erkennungsmechanismen.

Mit

`--heuristic-alerts=yes`

beziehungsweise

`--heuristic-alerts=no`

können heuristische Warnungen aktiviert beziehungsweise deaktiviert werden.

Standardmäßig sind heuristische Warnungen aktiviert.

Eine heuristische Meldung basiert nicht zwangsläufig auf derselben Art eindeutiger Signaturübereinstimmung wie ein klassischer Signaturfund.

Auch solche Ergebnisse müssen deshalb sachgerecht bewertet werden.

Verschlüsselte Archive und Dokumente

Mit

`--alert-encrypted`

kann ClamAV so konfiguriert werden, dass verschlüsselte Archive beziehungsweise Dokumente gemeldet werden.

Verschlüsselter Inhalt kann für einen Scanner problematisch sein, weil dessen eigentlicher Inhalt ohne Entschlüsselung nicht vollständig analysiert werden kann.

Eine entsprechende Meldung bedeutet jedoch nicht automatisch, dass die Datei Schadsoftware enthält.

Sie weist zunächst auf den verschlüsselten und damit nicht normal untersuchbaren Inhalt hin.

Temporäre Dateien

Mit

`--leave-temps`

kann verhindert werden, dass von ClamAV während der Verarbeitung erzeugte temporäre Dateien anschließend entfernt werden.

Diese Funktion ist primär für spezielle Analyse- und Debuggingzwecke relevant.

Im normalen Scanbetrieb besteht in der Regel kein Grund, temporäre Scanartefakte dauerhaft zurückzulassen.

clamscan und clamdscan

ClamAV stellt mehrere Scanwege bereit.

`clamscan` ist ein eigenständiger Scanner.

Er lädt bei jedem Start die Signaturdatenbank in eine neu erzeugte Scan-Engine und benötigt keinen laufenden `clamd`.

`clamdscan` ist dagegen ein Client für den bereits laufenden ClamAV-Daemon `clamd`.

Der Daemon hält seine Scan-Engine und Datenbank im Speicher und kann Scanaufträge von Clients entgegennehmen.

Für einen gezielten Einmalscan in einem Rescue-System ist `clamscan` besonders unkompliziert, weil kein zuvor konfigurierter und laufender Scan-Daemon erforderlich ist.

Kein vollständiges Endpoint-Security-System

ClamAV beschreibt sich als **Malware Detection Toolkit** und nicht als vollständige Endpoint-Security-Suite.

Ein manueller `clamscan` stellt insbesondere keinen permanenten Echtzeitschutz dar.

ClamAV besitzt zwar separate On-Access-Funktionen für Linux, diese arbeiten jedoch über `clamd` und `clamonacc` und sind funktional von einem einmaligen `clamscan` zu unterscheiden.

A.10 dokumentiert deshalb ausdrücklich den gezielten Rescue-Scan und nicht den Aufbau eines permanenten Endpoint-Schutzsystems.

Scan eines Windows-Datenträgers

Da `clamscan` Dateien und Verzeichnisse untersucht, kann ein unter Linux eingebundenes Dateisystem eines anderen Betriebssystems ebenfalls als Scanziel verwendet werden, soweit Linux auf dessen Dateien zugreifen kann.

Dies ist für ein Rescue-System besonders nützlich.

Ein Windows-System muss nicht selbst gestartet werden, um dessen zugängliche Dateien von JUMBO BLOCK Rescue aus mit ClamAV zu untersuchen.

Damit kann ein möglicherweise kompromittiertes installiertes Betriebssystem offline untersucht werden.

Vorteil eines extern gestarteten Rescue-Systems

Wird JUMBO BLOCK Rescue unabhängig vom installierten Zielbetriebssystem gestartet, läuft der Scan nicht innerhalb dieses möglicherweise kompromittierten Betriebssystems.

Die Dateien des Zielsystems können stattdessen von der Rescue-Umgebung aus untersucht werden.

Dies verhindert jedoch nicht automatisch sämtliche denkbaren Manipulations- oder Erkennungsprobleme und ersetzt keine vollständige forensische Untersuchung.

Es bietet aber einen klaren praktischen Vorteil:

Die Programme des zu untersuchenden Betriebssystems müssen für einen dateibasierten ClamAV-Scan nicht ausgeführt werden.

Fund ist nicht gleich Beweis

Ein ClamAV-Fund ist ein technisches Scanergebnis.

Er sollte dokumentiert und anschließend bewertet werden.

False Positives sind möglich.

Ebenso bedeutet ein Scan ohne Fund nicht, dass ein System mit absoluter Sicherheit frei von Schadsoftware ist.

Erkennung hängt unter anderem von den verwendeten Signaturen, der Scan-Engine, den Scanoptionen, Dateizugriffsrechten, unterstützten Formaten, Größenlimits und der grundsätzlichen Erkennbarkeit einer Bedrohung ab.

Zugriffsrechte

ClamAV kann nur Inhalte untersuchen, auf die der Scanprozess zugreifen kann.

Fehlende Leserechte können daher dazu führen, dass Dateien nicht untersucht werden.

Bei der Auswertung eines Scanberichts müssen Fehlermeldungen und nicht untersuchte Dateien berücksichtigt werden.

Ein Scan darf nicht allein anhand der Anzahl gemeldeter Funde beurteilt werden.

Scan und beschädigte Datenträger

ClamAV ist kein Werkzeug zur physischen Datenrettung.

Bei einem Laufwerk mit Lesefehlern oder anderen Hardwareproblemen sollte nicht zunächst ein umfangreicher Malware-Scan erzwungen werden.

Ein vollständiger Scan kann große Teile eines Datenträgers lesen und ihn entsprechend beanspruchen.

Wenn wichtige Daten auf einem instabilen Medium gerettet werden müssen, hat die Sicherung des noch lesbaren Inhalts – beispielsweise mit GNU ddrescue – Vorrang.

Anschließend kann ein erzeugtes Image beziehungsweise dessen zugänglicher Inhalt untersucht werden.

ClamAV und Datenrettung

Die Aufgaben müssen klar getrennt werden:

GNU ddrescue sichert möglichst viele lesbare Blöcke eines problematischen Mediums.

TestDisk untersucht und rekonstruiert Partitionierungs- beziehungsweise bestimmte Dateisystemstrukturen.

PhotoRec und Foremost suchen Dateien anhand ihrer Inhalte beziehungsweise Signaturen.

ClamAV Scan untersucht zugängliche Dateien auf bekannte beziehungsweise durch die ClamAV-Engine erkennbare schädliche oder verdächtige Inhalte.

ClamAV ersetzt damit kein Datenrettungswerkzeug.

Typischer Rescue-Ablauf

Bei einem technisch intakten, aber sicherheitsseitig verdächtigen Datenträger bietet sich ein kontrollierter Ablauf an.

Zunächst wird das richtige Dateisystem identifiziert und unter kontrollierten Bedingungen zugänglich gemacht.

Danach wird geprüft, ob die vorhandene ClamAV-Signaturdatenbank für den vorgesehenen Scan geeignet beziehungsweise aktuell genug ist.

Anschließend kann der relevante Verzeichnisbaum beispielsweise rekursiv untersucht werden:

```
clamscan -r /pfad/zum/ziel
```

Für eine übersichtlichere Fundausgabe:

```
clamscan -r -i /pfad/zum/ziel
```

Für eine dokumentierte Untersuchung kann zusätzlich ein Log geschrieben werden:

```
clamscan -r -i --log=/pfad/scan.log /pfad/zum/ziel
```

Gefundene Dateien sollten zunächst bewertet werden.

Eine automatische Löschung ist kein geeigneter Standardablauf.

Sicherheitshinweise

Vor einem Scan muss das richtige Ziel eindeutig identifiziert sein.

Bei Originalmedien mit wichtigen Daten sollte möglichst zunächst ein unveränderter beziehungsweise kontrolliert gesicherter Ausgangszustand erhalten werden.

Die Optionen

`--move`

und insbesondere

`--remove`

verändern den untersuchten Datenbestand.

Sie sind daher von einem rein diagnostischen Scan zu unterscheiden.

Ein fehlender Fund ist keine Garantie für Schadsoftwarefreiheit.

Ein Fund ist umgekehrt nicht automatisch der Beweis für eine tatsächlich aktive Infektion.

Scanfehler, Zugriffsprobleme und Scanlimits müssen bei der Interpretation des Ergebnisses berücksichtigt werden.

Rolle in JUMBO BLOCK Rescue

ClamAV Scan bildet innerhalb von JUMBO BLOCK Rescue die erste spezialisierte Ebene für die **dateibasierte Malware-Untersuchung**.

Seine besondere Stärke im Rescue-Kontext besteht darin, dass ein installiertes Zielbetriebssystem nicht gestartet werden muss.

Ein zugängliches Dateisystem kann aus der unabhängigen Rescue-Umgebung heraus untersucht werden.

Die funktionale Einordnung lautet:

ClamAV Scan / `clamscan` – einmalige Malware-Untersuchung von Dateien und Verzeichnissen

ClamAV Signaturen / `freshclam` – Aktualisierung der ClamAV-Erkennungsdatenbanken

Chkrootkit – Untersuchung auf Hinweise bestimmter Rootkits und verdächtiger Systemmerkmale

Rkhunter – zusätzliche hostbasierte Sicherheits- und Rootkit-Prüfungen

GUFW Firewall – grafische Verwaltung der hostbasierten Firewall

ClamAV Scan ist damit kein Reparaturwerkzeug und kein vollständiges Endpoint-Security-System.

Seine Aufgabe innerhalb von JUMBO BLOCK Rescue ist die **kontrollierte Untersuchung zugänglicher Dateien auf durch ClamAV erkennbare Bedrohungen**, vorzugsweise ohne das möglicherweise kompromittierte Zielbetriebssystem selbst starten zu müssen.

A.11 ClamAV Signaturen

Bereich: Sicherheit
Programm: ClamAV Signaturen
Technische Grundlage: ClamAV / `freshclam`
Baseline 20261001: ClamAV 1.4.3
Installierte Pakete: `clamav-freshclam 1.4.3+dfsg-1`, `clamav-base 1.4.3+dfsg-1`, `clamav 1.4.3+dfsg-1`
Bedienung: Kommandozeilenbasierte Aktualisierung der Malware-Signaturdatenbanken

Zweck

ClamAV Signaturen dient innerhalb von JUMBO BLOCK Rescue zur Pflege der Erkennungsdatenbanken, die vom ClamAV-Scanner für die Malware-Untersuchung verwendet werden.

Das zentrale Programm dafür ist:

`freshclam`

FreshClam ist das offizielle automatische Datenbank-Aktualisierungswerkzeug von ClamAV.

Es lädt die offiziellen ClamAV-Signaturdatenbanken herunter und aktualisiert bereits vorhandene Datenbanken.

Damit sind die Aufgaben von A.10 und A.11 klar getrennt:

A.10 – ClamAV Scan / `clamscan`
untersucht Dateien und Verzeichnisse.

A.11 – ClamAV Signaturen / `freshclam`
stellt die dafür benötigten offiziellen Erkennungsdatenbanken bereit und aktualisiert sie.

Bedeutung der Signaturdatenbanken

Bevor die ClamAV-Scan-Engine mit `clamscan` oder `clamd` verwendet werden kann, müssen ClamAV-Virendatenbanken am dafür vorgesehenen Ort vorhanden sein.

Diese Datenbanken stellen einen wesentlichen Teil der Erkennungsgrundlage des Scanners dar.

Die Version der installierten ClamAV-Software und der Aktualitätsstand der Signaturdatenbanken sind dabei zwei unterschiedliche Dinge.

Ein installiertes:

`ClamAV 1.4.3`

bedeutet daher nicht automatisch, dass auch die vorhandenen Signaturdatenbanken aktuell sind.

Für die Beurteilung eines Malware-Scans muss beides berücksichtigt werden.

FreshClam

FreshClam wird grundsätzlich mit

```
freshclam
```

aufgerufen.

Ein typischer manueller Aufruf lautet:

```
sudo freshclam
```

FreshClam benötigt eine gültige Konfigurationsdatei `freshclam.conf`.

Die konkrete Position dieser Datei hängt von Installation und Distribution ab.

ClamAV weist ausdrücklich darauf hin, dass paketierte Installationen die Konfiguration bereits distributionsspezifisch vorbereiten können.

Aktualisierungsablauf

Beim normalen Aufruf versucht FreshClam, eine Verbindung zum ClamAV-Verteilnetz für die Virensignaturdatenbanken herzustellen.

Existieren im konfigurierten Datenbankverzeichnis noch keine Datenbanken, lädt FreshClam die benötigten Datenbanken vollständig herunter.

Sind bereits Datenbanken vorhanden, versucht FreshClam diese zu aktualisieren.

Damit muss bei einer Aktualisierung nicht grundsätzlich jedes Mal die vollständige Datenbank neu heruntergeladen werden.

Inkrementelle Aktualisierung

FreshClam unterstützt sogenannte **CDIFF-Updates**.

Dabei werden Unterschiede zwischen dem vorhandenen und dem aktuellen Datenbankstand übertragen und auf die lokale Datenbank angewendet.

ClamAV beschreibt FreshClam ausdrücklich als Werkzeug, das anstelle der vollständigen CVD-Datei bei jeder Aktualisierung nur die Unterschiede zwischen vorhandener und aktueller Datenbank übertragen kann.

Dies reduziert die zu übertragende Datenmenge.

Beschädigte Datenbank

FreshClam prüft den Zustand vorhandener Datenbanken während des Aktualisierungsvorgangs.

Wird eine vorhandene Datenbank als beschädigt erkannt, wird sie nicht einfach inkrementell weitergeführt.

ClamAV dokumentiert, dass eine beschädigte Datenbank stattdessen durch einen vollständigen neuen Download ersetzt wird.

Damit besitzt FreshClam einen Mechanismus, um eine nicht mehr verwendbare lokale Datenbank wieder durch einen vollständigen Datenbankstand zu ersetzen.

Datenbanktypen

ClamAV verwendet verschiedene Datenbankdateien und Signaturformate.

Zu den zentralen offiziellen Datenbanken gehören die von ClamAV bereitgestellten Virusdatenbanken.

ClamAV bezeichnet die vollständigen signierten Datenbankcontainer als **CVD – ClamAV Virus Database**.

FreshClam verwaltet die von ClamAV veröffentlichten Datenbankstände automatisch.

Für den normalen JUMBO BLOCK Rescue Einsatz ist deshalb keine manuelle Bearbeitung der einzelnen Signaturdateien erforderlich.

CVD

CVD-Dateien sind signierte ClamAV-Datenbankarchive.

Sie enthalten Signaturdaten und Metadaten, die von der ClamAV-Engine verwendet werden können.

FreshClam übernimmt die Beschaffung beziehungsweise Aktualisierung dieser Datenbanken.

Eine manuelle Veränderung offizieller CVD-Dateien ist für den normalen Rescue-Betrieb nicht vorgesehen.

CDIFF

CDIFF-Dateien beziehungsweise CDIFF-Inhalte dienen FreshClam zur inkrementellen Aktualisierung.

Anstatt immer eine komplette neue Datenbank zu übertragen, können Änderungen gegenüber dem vorhandenen Datenbankstand angewendet werden.

Damit ist zwischen zwei Situationen zu unterscheiden:

Erstinstallation beziehungsweise fehlende Datenbank:
vollständiger Download.

Vorhandene aktualisierbare Datenbank:
FreshClam versucht eine inkrementelle Aktualisierung.

Ist die lokale Datenbank beschädigt, kann FreshClam wieder auf einen vollständigen Download zurückgreifen.

Digitale Signaturen

FreshClam unterstützt die Verifikation digitaler Signaturen der ClamAV-Datenbanken.

Die Integritäts- und Authentizitätsprüfung der offiziellen Datenbankdateien ist ein wesentlicher Bestandteil der sicheren Signaturversorgung.

Eine Malware-Datenbank sollte nicht aus einer beliebigen oder ungeprüften Quelle in ein Rescue-System übernommen werden.

Manuelle Aktualisierung

Für einen unmittelbar gewünschten Updateversuch kann FreshClam manuell gestartet werden:

```
sudo freshclam
```

Der Ablauf und eventuelle Fehlermeldungen werden dabei direkt im Terminal ausgegeben.

Dies ist für JUMBO BLOCK Rescue besonders nützlich, wenn vor einem Malware-Scan geprüft werden soll, ob die vorhandenen Signaturen aktualisiert werden können.

Interaktiver Betrieb

FreshClam kann **on demand**, also bei Bedarf über die Kommandozeile, ausgeführt werden.

Dies ist der für ein Rescue-System besonders naheliegende Betriebsmodus.

Der Anwender entscheidet bewusst, wann eine Internetverbindung hergestellt und eine Aktualisierung versucht wird.

Nach erfolgreicher Aktualisierung können anschließend die relevanten Datenträger mit `clamscan` untersucht werden.

Daemon-Modus

FreshClam unterstützt zusätzlich einen Hintergrundbetrieb.

Mit

```
freshclam -d
```

kann FreshClam im Daemon-Modus ausgeführt werden.

In diesem Modus kann das Programm regelmäßig nach neuen Datenbankständen suchen.

Dieser Betriebsmodus ist vor allem für dauerhaft laufende Systeme relevant.

Ein Rescue-System, das nur für einen konkreten Wartungs- oder Notfalleinsatz gestartet wird, benötigt nicht zwangsläufig einen permanent laufenden FreshClam-Daemon.

DNS-basierte Versionsprüfung

FreshClam unterstützt Datenbank-Versionsprüfungen über DNS.

Damit kann das Programm feststellen, ob neuere Datenbankstände verfügbar sind, ohne für jede Prüfung zwangsläufig die vollständige Datenbank übertragen zu müssen.

Diese Funktion gehört zum normalen Aktualisierungsmechanismus von FreshClam.

Proxy-Unterstützung

FreshClam kann über einen HTTP-Proxy arbeiten.

Auch Proxy-Authentifizierung wird unterstützt.

Die entsprechenden Einstellungen werden über `freshclam.conf` konfiguriert.

Dies kann relevant sein, wenn JUMBO BLOCK Rescue in einer Unternehmens- oder KRITIS-Umgebung betrieben wird, in der Internetzugriffe ausschließlich über einen Proxy erlaubt sind.

Die konkreten Proxydaten müssen der jeweiligen Umgebung entsprechen und dürfen nicht pauschal vorgegeben werden.

Konfigurationsdatei

FreshClam benötigt:

`freshclam.conf`

Die Konfiguration steuert unter anderem den Updatebetrieb.

ClamAV weist darauf hin, dass der genaue Speicherort installationsabhängig sein kann.

Bei paketierten Linux-Installationen wird die Konfiguration üblicherweise durch das Paket beziehungsweise die Distribution vorbereitet.

Deshalb sollte die tatsächlich installierte Konfiguration verwendet und nicht durch eine aus einer anderen Installation übernommene Beispielkonfiguration ersetzt werden.

Konfiguration prüfen

ClamAV stellt mit

`clamconf`

ein Werkzeug zur Untersuchung der ClamAV-Systemkonfiguration bereit.

Ein Aufruf von

`clamconf`

zeigt unter anderem wichtige Build-, Plattform-, Engine-, Datenbank- und Konfigurationsinformationen.

Dabei können auch `freshclam.conf` und andere ClamAV-Konfigurationsdateien berücksichtigt werden.

Dies ist hilfreich, wenn ein Signaturupdate aufgrund einer Konfigurationsabweichung nicht funktioniert.

Datenbankverzeichnis

FreshClam schreibt die heruntergeladenen Signaturdatenbanken in das konfigurierte ClamAV-Datenbankverzeichnis.

Der Benutzer, unter dem FreshClam ausgeführt wird, benötigt dort die notwendigen Schreibrechte.

Die Programme, welche die Datenbanken anschließend zum Scannen verwenden, benötigen mindestens ausreichenden Lesezugriff.

Der konkrete Pfad sollte aus der installierten Konfiguration beziehungsweise der ClamAV-Konfiguration des Systems ermittelt werden.

Er sollte im Rescue-Handbuch nicht ohne Prüfung als universeller Pfad vorausgesetzt werden.

Datenbankbesitz und Rechte

ClamAV unterstützt in `freshclam.conf` die Einstellung:

`DatabaseOwner`

Damit kann festgelegt werden, unter welchem Benutzerkontext die Datenbank verwaltet wird.

ClamAV weist darauf hin, dass dieser Benutzer Eigentümer des Datenbankverzeichnisses sein beziehungsweise die notwendigen Rechte besitzen muss, damit FreshClam Updates speichern kann.

Fehlende Schreibrechte können ein Signaturupdate verhindern.

Update-Protokoll

FreshClam kann seine Aktivitäten protokollieren.

Über die Konfiguration kann eine Update-Logdatei festgelegt werden.

Ein solches Protokoll kann insbesondere dann hilfreich sein, wenn Aktualisierungen fehlschlagen oder der zuletzt erfolgreiche Updatevorgang nachvollzogen werden soll.

Für ein Rescue-System kann dies bei der Fehlersuche nützlich sein.

Netzwerkverbindung

Für einen normalen Online-Updatevorgang benötigt FreshClam eine funktionierende Netzwerkverbindung zum ClamAV-Verteilnetz.

Ohne erreichbare Updatequelle kann FreshClam keine neuen offiziellen Signaturen herunterladen.

Dies bedeutet jedoch nicht automatisch, dass ClamAV überhaupt nicht scannen kann.

Sind bereits verwendbare lokale Datenbanken vorhanden, kann `clamscan` mit diesen vorhandenen Datenbanken arbeiten.

Der Scan erfolgt dann jedoch mit genau dem lokal vorhandenen Signaturstand.

Offline-Betrieb

JUMBO BLOCK Rescue kann auch in einer Umgebung eingesetzt werden, in der keine Internetverbindung verfügbar oder zulässig ist.

In diesem Fall kann FreshClam keinen normalen Online-Abgleich mit dem ClamAV-Verteilnetz durchführen.

ClamAV Scan kann vorhandene lokale Signaturdatenbanken weiterhin verwenden, sofern diese gültig und für die installierte Engine nutzbar sind.

Für die Bewertung des Scanergebnisses muss dann jedoch berücksichtigt werden, dass die Signaturen möglicherweise nicht dem aktuellen Veröffentlichungsstand entsprechen.

Offline bedeutet nicht wirkungslos

Ein Malware-Scan mit einer älteren, aber gültigen Signaturdatenbank ist nicht automatisch wirkungslos.

Die vorhandenen Erkennungsdaten bleiben grundsätzlich verfügbar.

Allerdings fehlen Signaturen und Datenbankänderungen, die erst nach dem letzten Aktualisierungsstand veröffentlicht wurden.

Deshalb muss ein Offline-Scan korrekt beschrieben werden als:

Scan mit dem vorhandenen lokalen Signaturstand.

Er darf nicht als Scan mit garantiert aktuellen Signaturen bezeichnet werden.

Aktualisierung vor einem Scan

Ist eine Netzwerkverbindung verfügbar und nach den Sicherheitsregeln des Einsatzortes zulässig, kann vor einem wichtigen Malware-Scan zunächst

```
sudo freshclam
```

ausgeführt werden.

Nach einem erfolgreichen Update steht dem anschließend gestarteten `clamscan` der aktualisierte lokale Datenbankstand zur Verfügung.

Ein sinnvoller Rescue-Ablauf lautet damit:

Signaturstand prüfen beziehungsweise aktualisieren,

danach Zielmedium identifizieren,

anschließend Malware-Scan durchführen,

und Ergebnisse dokumentieren.

Keine Internetverbindung erzwingen

In bestimmten Sicherheits-, KRITIS- oder forensischen Szenarien kann es ausdrücklich unerwünscht sein, das Rescue-System mit einem Netzwerk zu verbinden.

Die Aktualität der Malware-Signaturen muss gegen diese Sicherheitsanforderung abgewogen werden.

Ein Signaturupdate ist kein Grund, eine bewusst isolierte Umgebung ohne Freigabe mit einem externen Netzwerk zu verbinden.

In einem solchen Fall wird mit dem vorhandenen lokalen Datenbankstand gearbeitet oder ein kontrolliertes Offline-Updateverfahren der jeweiligen Organisation verwendet.

Kontrolliertes Offline-Update

ClamAV-Datenbanken können grundsätzlich auch lokal bereitgestellt werden.

Für ein kontrolliertes Offline-Szenario müssen die benötigten Datenbankdateien über einen vertrauenswürdigen Prozess auf das Rescue-System beziehungsweise in das vorgesehene Datenbankverzeichnis übertragen werden.

Dabei sind mindestens zu berücksichtigen:

Herkunft der Datenbanken,

Integrität,

Kompatibilität mit der verwendeten ClamAV-Version,

Dateirechte,

und der tatsächlich verwendete Datenbankpfad.

Ein organisationsspezifisches Offline-Updateverfahren ist nicht Bestandteil der Baseline-Paketliste und darf daher nicht als bereits implementierte JUMBO-BLOCK-Funktion vorausgesetzt werden.

FreshClam und lokale Mirrors

FreshClam kann über seine Konfiguration unterschiedliche Update-Infrastrukturen verwenden.

In größeren oder abgeschotteten Organisationen kann die Signaturversorgung deshalb grundsätzlich über eine kontrollierte interne Infrastruktur organisiert werden.

Ob eine solche Infrastruktur vorhanden ist, ist jedoch installations- beziehungsweise organisationsabhängig.

JUMBO BLOCK Rescue setzt keinen solchen Server voraus.

Signaturen und Engine-Kompatibilität

Signaturdatenbanken und Scan-Engine müssen miteinander kompatibel sein.

Neue Datenbankfunktionen können bestimmte Mindestfähigkeiten der Engine voraussetzen.

Deshalb darf die Signaturversorgung nicht vollständig losgelöst von der installierten ClamAV-Version betrachtet werden.

Für die Baseline 20261001 ist die installierte Engine:

ClamAV 1.4.3

maßgebend.

Fehler beim Update

Ein fehlgeschlagenes FreshClam-Update kann unterschiedliche Ursachen besitzen.

Dazu können unter anderem gehören:

fehlende Netzwerkverbindung,

nicht erreichbare Updateinfrastruktur,

fehlerhafte Konfiguration,

Proxyprobleme,

fehlende Schreibrechte,

oder Probleme mit der lokalen Datenbank.

Die konkrete Fehlermeldung von FreshClam sollte deshalb ausgewertet werden, anstatt einen fehlgeschlagenen Updateversuch pauschal als Defekt von ClamAV zu interpretieren.

Wiederholte Updateversuche

FreshClam sollte nicht in sehr kurzen Abständen unkontrolliert wiederholt gestartet werden, wenn eine Aktualisierung aufgrund eines Netzwerk- oder Serverproblems fehlschlägt.

Ein fehlgeschlagenes Update sollte zunächst anhand der ausgegebenen Meldung analysiert werden.

Insbesondere Netzwerk-, DNS-, Proxy- oder Konfigurationsprobleme werden durch bloßes wiederholtes Starten des gleichen Befehls nicht behoben.

Signaturen sind keine Garantie

Auch vollständig aktuelle ClamAV-Signaturen garantieren nicht, dass jede existierende Schadsoftware erkannt wird.

Signaturbasierte und weitere von ClamAV verwendete Erkennungsverfahren können nur Bedrohungen erkennen, für die entsprechende Erkennungsmerkmale vorhanden beziehungsweise durch die Engine erfassbar sind.

Umgekehrt können False Positives auftreten.

Die Signaturaktualisierung erhöht deshalb die Qualität und Aktualität der Erkennungsbasis, ersetzt aber keine umfassende Sicherheitsanalyse.

Signaturstand und Scanbericht

Bei einer professionellen Untersuchung sollte neben dem eigentlichen Scanergebnis auch berücksichtigt werden, mit welchem Datenbankstand der Scan durchgeführt wurde.

Dies erhöht die Nachvollziehbarkeit.

Ein späterer Scan mit aktualisierten Signaturen kann zu einem anderen Ergebnis kommen als ein früherer Scan derselben Dateien.

Das ist insbesondere bei neu hinzugekommenen Erkennungen möglich.

Trennung von Update und Scan

FreshClam untersucht selbst keine Zielverzeichnisse auf Malware.

Seine Aufgabe ist die Pflege der Erkennungsdatenbanken.

Umgekehrt ist `clamscan` nicht das Werkzeug zur regulären Aktualisierung dieser Datenbanken.

Damit bleibt die funktionale Trennung eindeutig:

`freshclam`
Signaturversorgung

`clamscan`
Malware-Scan

Diese Trennung wird in JUMBO BLOCK Rescue bewusst durch zwei getrennte Handbuchkapitel abgebildet.

Typischer Rescue-Ablauf mit Internetverbindung

Ist eine Internetverbindung erlaubt und verfügbar, kann der Ablauf beispielsweise so aussehen:

```
sudo freshclam
```

Anschließend wird geprüft, ob die Aktualisierung erfolgreich abgeschlossen wurde.

Danach kann der relevante Datenträger mit dem in A.10 beschriebenen `clamscan` untersucht werden.

Das Update sollte vor dem Scan erfolgen, damit dieser mit dem aktuell lokal verfügbaren Datenbankstand arbeitet.

Typischer Rescue-Ablauf ohne Internetverbindung

Ist keine Internetverbindung vorhanden oder erlaubt, wird kein erfolgreicher Online-Updatevorgang vorausgesetzt.

Stattdessen wird mit den bereits lokal vorhandenen, gültigen Signaturdatenbanken gearbeitet.

Der resultierende Scan ist dann ausdrücklich ein:

Scan mit dem auf JUMBO BLOCK Rescue vorhandenen Signaturstand.

Die zeitliche Aktualität dieses Stands muss bei der Interpretation berücksichtigt werden.

Sicherheitshinweise

Signaturdatenbanken sollten nur aus vertrauenswürdigen Quellen bezogen werden.

Ein Rescue-System sollte nicht allein für ein Signaturupdate ungeprüft mit einem Netzwerk verbunden werden, wenn der Einsatz eine isolierte Umgebung verlangt.

Ein erfolgreiches FreshClam-Update bedeutet nicht, dass anschließend jede Schadsoftware erkannt werden kann.

Ein fehlgeschlagenes Update bedeutet umgekehrt nicht zwangsläufig, dass ClamAV nicht mehr scannen kann.

Entscheidend ist, ob eine gültige lokale Datenbank vorhanden ist.

Bei Offline-Einsätzen muss der vorhandene Signaturstand als solcher dokumentiert und bei der Bewertung des Scanergebnisses berücksichtigt werden.

Rolle in JUMBO BLOCK Rescue

ClamAV Signaturen bildet innerhalb von JUMBO BLOCK Rescue die **Versorgungsebene für die Malware-Erkennungsdatenbanken**.

Die funktionale Kette lautet:

FreshClam → Signaturdatenbanken aktualisieren

ClamScan → Dateien mit den lokal vorhandenen Signaturdatenbanken untersuchen

Damit ergänzen sich A.10 und A.11 unmittelbar.

ClamAV Scan beantwortet die Frage:

„Was wird in den ausgewählten Dateien erkannt?“

ClamAV Signaturen beantwortet die vorgelagerte Frage:

„Mit welchem Erkennungsdatenbestand arbeitet der Scanner?“

Gerade bei einem portablen Rescue-System ist diese Unterscheidung wesentlich, weil JUMBO BLOCK Rescue sowohl online mit zuvor aktualisierten Signaturen als auch isoliert mit dem vorhandenen lokalen Datenbankstand eingesetzt werden kann.

Der Signaturstand ist deshalb ein wesentlicher Bestandteil der technischen Aussagekraft eines ClamAV-Scans.

A.12 Chkrootkit

Bereich: Sicherheit
Programm: Chkrootkit
Baseline 20261001: Chkrootkit 0.58b
Installiertes Paket: `chkrootkit 0.58b-5+b8`
Bedienung: Kommandozeile
Aufgabe: Lokale Untersuchung eines Unix-/Linux-Systems auf Anzeichen von Rootkits und Manipulationen

Zweck

Chkrootkit dient zur lokalen Untersuchung eines Systems auf **Anzeichen von Rootkits und bestimmte Spuren einer Systemkompromittierung**.

Der Name steht für:

Check Rootkit

Die JUMBO BLOCK Rescue Baseline 20261001 enthält:

`chkrootkit 0.58b-5+b8`

Chkrootkit ist kein klassischer Virenscanner wie ClamAV.

Statt Dateien allgemein auf Malware-Signaturen zu untersuchen, führt Chkrootkit eine Reihe spezialisierter Prüfungen durch, die auf typische Manipulationen, Dateien, Prozesse, Netzwerkzustände und andere Merkmale bestimmter Rootkits beziehungsweise kompromittierter Systeme abzielen.

Was ist ein Rootkit?

Als Rootkit wird Software beziehungsweise eine Sammlung von Komponenten bezeichnet, die nach einer Kompromittierung eines Systems dazu eingesetzt werden kann, privilegierten Zugriff aufrechtzuerhalten oder Aktivitäten und Bestandteile eines Angriffs zu verbergen.

Rootkits können beispielsweise versuchen, Prozesse, Dateien, Netzwerkverbindungen oder Login-Spuren vor dem Administrator zu verbergen.

Chkrootkit sucht nach bestimmten bekannten Merkmalen und Inkonsistenzen, die auf solche Manipulationen hinweisen können.

Bestandteile von Chkrootkit

Chkrootkit besteht nicht nur aus einem einzelnen Test.

Das Projekt umfasst mehrere Komponenten.

Der zentrale Bestandteil ist:

`chkrootkit`

Ein Shell-Skript, das Systemprogramme und andere Systembestandteile auf Hinweise einer Rootkit-Manipulation untersucht.

Weitere vom Projekt dokumentierte Komponenten sind unter anderem:

`ifpromisc`
prüft Netzwerkinterfaces auf Promiscuous Mode.

`chklastlog`
prüft auf gelöschte beziehungsweise auffällige Einträge im Zusammenhang mit `lastlog`.

`chkwtmp`
prüft `wtmp` auf Hinweise auf gelöschte Einträge.

`chkproc`
sucht nach Anzeichen bestimmter LKM-Trojaner beziehungsweise versteckter Prozesse.

`chkdirs`
sucht nach weiteren Anzeichen bestimmter LKM-Manipulationen.

`chkutmp`
prüft auf gelöschte beziehungsweise auffällige `utmp`-Einträge.

Die einzelnen Komponenten werden durch die Hauptprüfung entsprechend ihrer Aufgabe eingesetzt.

Grundaufruf

Der grundlegende Aufruf lautet:

```
sudo chkrootkit
```

Ohne Angabe eines einzelnen Tests führt Chkrootkit seine vorgesehenen Prüfungen gegen das Zielsystem durch.

Für eine umfassende Systemprüfung sind erhöhte Rechte zweckmäßig, da bestimmte Systembereiche andernfalls nicht vollständig zugänglich sein können.

Ausgaben

Chkrootkit gibt die Ergebnisse seiner einzelnen Prüfungen im Terminal aus.

Dabei können unterschiedliche Zustände beziehungsweise Meldungen erscheinen.

Entscheidend ist:

Eine auffällige Chkrootkit-Meldung ist zunächst ein Untersuchungshinweis.

Sie muss im Kontext des Systems bewertet werden.

Chkrootkit selbst beziehungsweise die Debian-Integration berücksichtigt ausdrücklich das Problem möglicher False Positives und stellt Möglichkeiten bereit, bekannte Fehlalarme von der Ausgabe bestimmter Tests auszunehmen.

Verfügbare Tests anzeigen

Mit

```
chkrootkit -l
```

werden die verfügbaren Tests aufgelistet.

Damit lässt sich feststellen, welche einzelnen Prüfungen von der installierten Chkrootkit-Version angeboten werden.

Die Tests umfassen sowohl allgemeine Prüfungen als auch Untersuchungen bestimmter Systemprogramme und bekannter Rootkit-Merkmale.

Einzelnen Test ausführen

Chkrootkit erlaubt die Angabe eines Testnamens hinter den Optionen.

Grundsyntax:

```
chkrootkit [OPTIONEN] [TEST]
```

Damit muss nicht zwangsläufig bei jeder Untersuchung der komplette Testsatz ausgeführt werden.

Welche Testnamen tatsächlich zur Verfügung stehen, kann mit

```
chkrootkit -l
```

ermittelt werden.

Quiet-Modus

Mit

```
-q
```

wird Chkrootkit im Quiet-Modus ausgeführt.

Beispiel:

```
sudo chkrootkit -q
```

In diesem Modus wird die Ausgabe von Tests unterdrückt, die nichts Verdächtiges gefunden haben.

Dadurch kann die Ausgabe bei einer Untersuchung deutlich kompakter werden.

Der Quiet-Modus bedeutet jedoch nicht, dass weniger Tests durchgeführt werden.

Er verändert primär die Darstellung der Ergebnisse.

Expert-Modus

Mit

```
-x
```

wird der Expert-Modus aktiviert.

Beispiel:

```
sudo chkrootkit -x
```

Dieser Modus erzeugt bei vielen Prüfungen zusätzliche Informationen darüber, was der jeweilige Test gefunden hat.

Die Ausgabe kann dadurch erheblich umfangreicher werden.

Der Expert-Modus eignet sich insbesondere zur genaueren Analyse einer zuvor auffällig gewordenen Prüfung.

Debug-Modus

Mit

```
-d
```

wird der Debug-Modus aktiviert.

Dieser zeigt detailliert, was Chkrootkit während seiner Ausführung durchführt.

Der Modus ist primär für Fehlersuche und technische Analyse gedacht.

Für eine normale erste Rootkit-Prüfung ist er nicht erforderlich.

Optionen nicht zusammenziehen

Bei Chkrootkit ist zu beachten, dass Optionen nicht wie bei vielen anderen Unix-Programmen kombiniert werden sollen.

Statt beispielsweise:

```
-qn
```

müssen sie getrennt angegeben werden:

```
-q -n
```

Dies entspricht der dokumentierten Bedienung von Chkrootkit.

Externes System untersuchen

Für JUMBO BLOCK Rescue ist eine Option besonders wichtig:

```
-r DIR
```

Damit kann ein anderes Verzeichnis als Wurzelverzeichnis des zu untersuchenden Systems verwendet werden.

Beispielprinzip:

```
sudo chkrootkit -r /mnt/zielsystem
```

Der konkrete Mountpunkt `/mnt/zielsystem` ist hierbei lediglich ein Beispiel.

Er muss durch den tatsächlichen Mountpunkt des zu untersuchenden Systems ersetzt werden.

Diese Funktion ist für ein Rescue-System besonders wertvoll.

Ein möglicherweise kompromittiertes Linux-System kann auf diese Weise von JUMBO BLOCK Rescue aus eingehängt und untersucht werden, ohne das installierte Zielsystem selbst starten zu müssen.

Vorteil der externen Untersuchung

Ein Rootkit kann versuchen, die Sicht des laufenden kompromittierten Systems zu manipulieren.

Wird Chkrootkit innerhalb genau dieses Systems ausgeführt, muss deshalb berücksichtigt werden, dass die zur Untersuchung verwendete Umgebung selbst kompromittiert sein könnte.

Die Option

`-r`

ermöglicht einen anderen Ansatz:

Das verdächtige Dateisystem wird unter einer unabhängigen Umgebung eingehängt und anschließend als Zielwurzel untersucht.

Genau diese Arbeitsweise passt zum Konzept von JUMBO BLOCK Rescue.

Das Zielbetriebssystem muss für eine solche dateisystembasierte Untersuchung nicht gestartet werden.

Alternative vertrauenswürdige Programme verwenden

Eine weitere sicherheitsrelevante Option ist:

`-p DIR1:DIR2:...`

Damit kann ein alternativer Suchpfad für externe Programme festgelegt werden, die Chkrootkit für seine Prüfungen benötigt.

Chkrootkit verwendet selbst Standardprogramme wie beispielsweise `find` und `grep`.

Das Werkzeug muss daher grundsätzlich berücksichtigen, dass solche Programme auf einem kompromittierten System manipuliert worden sein könnten.

Mit `-p` kann stattdessen ein Pfad zu vertrauenswürdigen Kopien dieser Programme vorgegeben werden.

Beispielprinzip:

```
sudo chkrootkit -p /media/usb
```

Die Debian-Dokumentation nennt genau diesen Einsatzzweck: vertrauenswürdige Kopien der benötigten Programme können an einem Ort abgelegt werden, an dem sie vom untersuchten System nicht verändert werden können.

Bedeutung für JUMBO BLOCK Rescue

Gerade diese Arbeitsweise verdeutlicht einen zentralen Vorteil eines unabhängigen Rescue-Systems.

Wird ein verdächtiges installiertes System von JUMBO BLOCK Rescue aus untersucht, stehen die Programme der Rescue-Umgebung unabhängig vom möglicherweise manipulierten Zielbetriebssystem zur Verfügung.

Trotzdem muss bei jeder Untersuchung genau geprüft werden, welche Programme und Daten Chkrootkit tatsächlich verwendet.

Eine externe Rescue-Umgebung macht eine Untersuchung robuster, ersetzt aber keine vollständige forensische Analyse.

NFS-Verzeichnisse überspringen

Mit

`-n`

können bestimmte Prüfungen angewiesen werden, NFS-gemountete Verzeichnisse zu ignorieren.

Dies kann verhindern, dass Netzwerkdateisysteme unnötig in eine lokale Untersuchung einbezogen werden.

Dateisystemtypen ausschließen

Die Debian-Integration von Chkrootkit unterstützt außerdem:

`-T FSTYPE`

Damit können bei bestimmten Prüfungen Dateisysteme eines angegebenen Typs ausgeschlossen werden.

Dies basiert auf der entsprechenden Dateisystemtyp-Prüfung von `find`.

Für einen gezielten Rescue-Scan kann diese Möglichkeit helfen, nicht relevante Dateisysteme von bestimmten Suchläufen auszunehmen.

Bekannte False Positives ausschließen

Chkrootkit unterstützt:

`-e "DATEI1 DATEI2 ..."`

Damit können bekannte Dateien beziehungsweise Verzeichnisse aus den Ergebnissen bestimmter Tests ausgeschlossen werden.

Die Liste wird leerzeichengetrennt angegeben und muss deshalb in der Shell üblicherweise in Anführungszeichen gesetzt werden.

Die Option kann auch mehrfach verwendet werden.

Wichtig ist:

`-e` sollte nicht dazu dienen, eine unbekannte Warnung einfach verschwinden zu lassen.

Eine Ausnahme sollte erst eingerichtet werden, wenn ein gemeldeter Fund tatsächlich untersucht und als erwarteter beziehungsweise bekannter False Positive identifiziert wurde.

Sniffer-Test und False Positives

Chkrootkit enthält einen Test auf Netzwerkinterfaces beziehungsweise Prozesse, die auf Packet-Sniffing hindeuten können.

Dabei können auch legitime Netzwerkprogramme auffallen.

Die Debian-Dokumentation nennt beispielsweise reguläre Netzwerkmanager wie:

`systemd-networkd`

`NetworkManager`

`wpa_supplicant`

als mögliche Auslöser entsprechender Meldungen.

Das ist ein wichtiges Beispiel dafür, warum ein Chkrootkit-Hinweis nicht automatisch eine Kompromittierung beweist.

Sniffer-Ausnahmen

Für den Sniffer-Test unterstützt Chkrootkit die Option:

`-s REGEXP`

Damit können erwartete Ergebnisse anhand eines regulären Ausdrucks aus der Ausgabe dieses Tests herausgefiltert werden.

Auch hier gilt:

Eine Ausnahme sollte nur für einen technisch nachvollzogenen legitimen Prozess eingerichtet werden.

Unbekannte Sniffer-Meldungen sollten zunächst untersucht werden.

Promiscuous Mode

Die Komponente `ifpromisc` prüft, ob Netzwerkinterfaces im **Promiscuous Mode** arbeiten.

Dieser Modus ermöglicht einem Interface grundsätzlich, mehr Netzwerkframes entgegenzunehmen als ausschließlich diejenigen, die direkt an dieses Interface adressiert sind.

Packet-Sniffer können diesen Modus verwenden.

Promiscuous Mode ist jedoch nicht automatisch ein Beweis für Schadsoftware.

Legitime Netzwerkdiagnose-, Monitoring- und Virtualisierungslösungen können ebenfalls entsprechende Betriebsarten verwenden.

Das Ergebnis muss deshalb im Kontext des untersuchten Systems bewertet werden.

Prüfung von Systemprogrammen

Ein zentraler Bestandteil von Chkrootkit besteht darin, Systemprogramme auf bekannte Hinweise von Rootkit-Manipulationen zu untersuchen.

Die Liste der verfügbaren Tests enthält zahlreiche typische Unix-/Linux-Programme und Dienste.

Dazu gehören beispielsweise Prüfungen im Umfeld von:

`login`

`ls`

`ps`

`netstat`

`passwd`

`sshd`

`syslogd`

`tcpdump`

`top`

und weiteren Systemkomponenten.

Dabei sucht Chkrootkit nach Merkmalen, die aus bekannten Rootkit-Manipulationen beziehungsweise bestimmten kompromittierten Versionen bekannt sind.

LKM-Prüfungen

Chkrootkit besitzt Komponenten zur Suche nach Anzeichen bestimmter **Loadable Kernel Module (LKM)**-basierter Trojaner beziehungsweise Rootkits.

Dazu gehören unter anderem:

`chkproc`

und

`chkdirs`.

Diese Prüfungen versuchen bestimmte Inkonsistenzen beziehungsweise Hinweise zu erkennen, die mit versteckten Prozessen oder Verzeichnissen in Verbindung stehen können.

Auch solche Ergebnisse sind als Indikatoren zu verstehen und müssen technisch überprüft werden.

Login-Spuren

Rootkits beziehungsweise Angreifer können versuchen, Spuren von Anmeldungen aus Systemprotokollen zu entfernen.

Chkrootkit besitzt deshalb Prüfprogramme für entsprechende Login-Datenbanken.

Dazu gehören unter anderem:

`chklastlog`

`chkwtmp`

`chkutmp`

Diese suchen nach bestimmten Auffälligkeiten beziehungsweise Hinweisen auf gelöschte Einträge.

chkwtmp

`chkwtmp` untersucht die Datei:

`/var/log/wtmp`

auf Einträge, die überschrieben beziehungsweise mit Nullbytes ersetzt wurden.

Werden entsprechende Bereiche gefunden, kann das Werkzeug anhand benachbarter Zeitstempel einen zeitlichen Bereich für den auffälligen Eintrag ausgeben.

Die Dokumentation nennt jedoch ausdrücklich technische Einschränkungen dieser Prüfung.

Ein auffälliges Ergebnis ist deshalb ein Hinweis für eine weitere Untersuchung und kein alleiniger Nachweis eines Angriffs.

chklastlog

`chklastlog` vergleicht Login-Informationen und sucht nach Benutzern, deren Anmeldeaktivität nicht konsistent in den dafür vorgesehenen Login-Daten erscheint.

Eine solche Inkonsistenz kann darauf hindeuten, dass Login-Spuren entfernt wurden.

Auch diese Prüfung besitzt dokumentierte Einschränkungen und muss im Kontext des jeweiligen Systems interpretiert werden.

Bekannte Rootkit-Merkmale

Chkrootkit enthält Prüfungen auf Merkmale zahlreicher bekannter Rootkits, Würmer und Trojaner.

Die Herstellerseite führt eine umfangreiche Liste der erkannten beziehungsweise geprüften Rootkit-Familien und Varianten.

Das bedeutet jedoch nicht, dass Chkrootkit jede existierende oder zukünftige Rootkit-Technik erkennen kann.

Die Erkennung basiert auf den implementierten Tests und bekannten Merkmalen.

Chkrootkit ist kein Virenscanner

Chkrootkit und ClamAV erfüllen unterschiedliche Aufgaben.

ClamAV

untersucht Dateien mit seiner Malware-Erkennungsengine und den geladenen Signaturdatenbanken.

Chkrootkit

untersucht das Unix-/Linux-System auf bestimmte Rootkit-Merkmale, Manipulationen und Inkonsistenzen.

Beide Werkzeuge können deshalb innerhalb einer Sicherheitsuntersuchung sinnvoll sein, ersetzen einander aber nicht.

Keine automatische Bereinigung

Chkrootkit ist ein **Erkennungs- und Analysewerkzeug**.

Seine Hauptaufgabe besteht darin, verdächtige Zustände festzustellen und zu melden.

Es ist kein allgemeines Werkzeug zur automatischen Entfernung gefundener Rootkits.

Eine bestätigte Systemkompromittierung erfordert eine separate Bewertung und geeignete Reaktionsmaßnahmen.

False Positives

False Positives sind bei Chkrootkit ausdrücklich zu berücksichtigen.

Legitime Dateien, Netzwerkdienste, Monitoringprogramme oder andere Systemkonfigurationen können bestimmte Tests auslösen.

Die Debian-Integration enthält deshalb sogar eine eigene Dokumentation bekannter False Positives und unterstützt Ausschlussoptionen.

Eine Meldung sollte daher zunächst als:

„prüfungsbedürftiger Befund“

behandelt werden.

Erst die anschließende technische Analyse entscheidet, ob tatsächlich eine Manipulation vorliegt.

Kein Fund ist ebenfalls kein Beweis

Auch die Meldung, dass Chkrootkit keine verdächtigen Merkmale gefunden hat, beweist nicht, dass das System garantiert kompromittierungsfrei ist.

Chkrootkit kann nur die Merkmale untersuchen, für die entsprechende Prüfverfahren implementiert sind und auf die es Zugriff besitzt.

Neue, unbekannte oder technisch anders arbeitende Rootkits können außerhalb dieser Erkennungslogik liegen.

Vertrauensproblem eines kompromittierten Systems

Bei Rootkit-Untersuchungen besteht ein grundsätzliches Problem:

Wurde das Betriebssystem selbst kompromittiert, können auch Programme manipuliert sein, die für die Untersuchung verwendet werden.

Genau deshalb unterstützt Chkrootkit mit `-p` alternative vertrauenswürdige Programme und mit `-r` die Untersuchung eines extern eingehängten Zielsystems.

Für JUMBO BLOCK Rescue ist dies ein zentraler konzeptioneller Vorteil.

Die Untersuchung kann aus einer separaten Rescue-Umgebung erfolgen.

Typischer Rescue-Ablauf

Ein sinnvoller erster Test der laufenden Umgebung lautet:

```
sudo chkrootkit
```

Für eine kompaktere Ausgabe:

```
sudo chkrootkit -q
```

Bei einer auffälligen Prüfung kann anschließend der betreffende Test gezielt beziehungsweise im Expert-Modus weiter untersucht werden.

Für ein extern eingehängtes verdächtiges Linux-System kann grundsätzlich:

```
sudo chkrootkit -r /pfad/zum/zielsystem
```

verwendet werden.

Der tatsächliche Mountpunkt muss vorher eindeutig festgestellt werden.

Vorgehen bei einer Warnung

Eine Chkrootkit-Warnung sollte nicht unmittelbar zu Änderungen am Originalsystem führen.

Stattdessen sollte zunächst festgestellt werden:

Welcher Test hat die Warnung ausgelöst?

Welche Datei, welcher Prozess oder welcher Systemzustand wurde beanstandet?

Ist der betreffende Zustand auf dem System erwartbar?

Ist für diesen Test ein bekannter False Positive dokumentiert?

Lässt sich der Befund mit anderen Werkzeugen unabhängig bestätigen?

Erst danach sollte über weitere Maßnahmen entschieden werden.

Beweissicherung

Besteht ein ernsthafter Verdacht auf eine Kompromittierung, sollte berücksichtigt werden, dass Änderungen am Originalsystem potenziell relevante Spuren verändern können.

In einem forensisch relevanten Fall ist deshalb die Sicherung des Ausgangszustands wichtiger als eine sofortige „Bereinigung“.

JUMBO BLOCK Rescue stellt mit seinen Datenrettungs- und Analysewerkzeugen Möglichkeiten bereit, Datenträger beziehungsweise Daten zunächst zu sichern.

Chkrootkit selbst ist jedoch kein forensisches Imaging-Werkzeug.

Chkrootkit auf einem beschädigten Datenträger

Bei einem physisch instabilen Datenträger muss zwischen Sicherheitsanalyse und Datenrettung unterschieden werden.

Chkrootkit benötigt Zugriff auf zahlreiche Dateien und Systembereiche.

Eine umfangreiche Untersuchung erzeugt daher Lesezugriffe.

Zeigt ein Laufwerk bereits Hardwarefehler und sind wichtige Daten zu retten, sollte nicht zunächst eine vollständige Rootkit-Untersuchung erzwungen werden.

In diesem Fall hat die Sicherung der noch lesbaren Daten Vorrang.

Chkrootkit und ClamAV

Die beiden vorherigen Kapitel und Chkrootkit bilden unterschiedliche Untersuchungsebenen:

ClamAV Signaturen
stellt die Malware-Erkennungsdatenbanken bereit.

ClamAV Scan
untersucht Dateien auf durch ClamAV erkennbare Malware.

Chkrootkit
sucht auf Unix-/Linux-Systemen nach bestimmten Anzeichen von Rootkits und Systemmanipulationen.

Eine umfassendere Sicherheitsuntersuchung kann deshalb mehrere dieser Werkzeuge kombinieren.

Grenzen

Chkrootkit besitzt klare Grenzen.

Es erkennt nicht jede mögliche Kompromittierung.

Es ersetzt keine vollständige forensische Untersuchung.

Es beweist anhand einer einzelnen Warnung nicht automatisch das Vorhandensein eines Rootkits.

Es kann False Positives erzeugen.

Und wenn es innerhalb eines bereits kompromittierten laufenden Systems verwendet wird, muss berücksichtigt werden, dass Teile der Untersuchungsumgebung manipuliert sein könnten.

Sicherheitshinweise

Chkrootkit-Ergebnisse müssen interpretiert werden.

Eine Meldung sollte nicht allein aufgrund ihres Wortlauts als bestätigte Infektion behandelt werden.

Bekannte False Positives dürfen erst nach technischer Prüfung ausgeblendet werden.

Bei einer Untersuchung eines fremden Systems sollte der richtige Mountpunkt eindeutig bestimmt werden.

Bei ernsthaftem Kompromittierungsverdacht sollten unnötige Änderungen am Originalsystem vermieden werden.

Bei physisch beschädigten Datenträgern hat die Datensicherung Vorrang vor umfangreichen Sicherheitsscans.

Rolle in JUMBO BLOCK Rescue

Chkrootkit bildet innerhalb von JUMBO BLOCK Rescue eine spezialisierte Untersuchungsebene für **Rootkit- und Manipulationshinweise auf Unix-/Linux-Systemen**.

Besonders relevant ist dabei die Möglichkeit, ein verdächtiges Linux-System von einer unabhängigen Rescue-Umgebung aus zu untersuchen.

Die Sicherheitswerkzeuge ergänzen sich damit:

ClamAV Scan – dateibasierte Malware-Untersuchung

ClamAV Signaturen – Pflege der ClamAV-Erkennungsdatenbanken

Chkrootkit – Suche nach Rootkit- und Manipulationshinweisen

Rkhunter – zusätzliche hostbasierte Rootkit- und Sicherheitsprüfung

GUFW Firewall – Verwaltung der hostbasierten Firewall

Chkrootkit liefert dabei keine automatische Entscheidung „System sicher“ oder „System kompromittiert“.

Seine Aufgabe ist präziser:

Es sucht nach technisch definierten Anzeichen, die eine weitergehende Untersuchung rechtfertigen können.

Genau in dieser Funktion ist Chkrootkit ein Diagnosewerkzeug innerhalb von JUMBO BLOCK Rescue.

A.13 Rkhunter

Bereich: Sicherheit
Programm: Rootkit Hunter (Rkhunter)
Baseline 20261001: Rkhunter 1.4.6
Installiertes Paket: `rkhunter 1.4.6-13`
Bedienung: Kommandozeile
Aufgabe: Untersuchung lokaler Linux-/Unix-Systeme auf Hinweise auf Rootkits, Malware und verdächtige Systemveränderungen

Zweck

Rkhunter steht für:

Rootkit Hunter

Das Programm führt verschiedene Sicherheitsprüfungen auf einem lokalen Linux-/Unix-System durch. Es versucht dabei, bekannte Rootkits und Malware sowie bestimmte verdächtige Veränderungen und Systemzustände zu erkennen.

Die JUMBO BLOCK Rescue Baseline 20261001 enthält:

`rkhunter 1.4.6-13`

Die zugrunde liegende Upstream-Version ist damit Rkhunter 1.4.6.

Rkhunter ist kein klassischer Virenscanner und ersetzt ClamAV nicht.

Das Programm kombiniert mehrere Arten von Sicherheitsprüfungen. Dazu gehören unter anderem Untersuchungen auf bekannte Rootkit-Merkmale, Veränderungen wichtiger Systemkommandos und Startdateien sowie Prüfungen von Netzwerkinterfaces und lauschenden Anwendungen.

Grundlegendes Prüfprinzip

Rkhunter arbeitet nicht mit nur einem einzigen Rootkit-Test.

Das Programm führt eine Reihe unterschiedlicher Prüfungen durch und fasst deren Ergebnisse zusammen.

Dazu gehören insbesondere:

Prüfungen von Systemkommandos,

Prüfungen gespeicherter Dateieigenschaften,

Suche nach bekannten Rootkit-Dateien und -Verzeichnissen,

Suche nach verdächtigen Dateien,

Prüfungen bestimmter Malware-Merkmale,

Prüfungen von Systemstartdateien,

Prüfungen von Netzwerkzuständen,

sowie weitere systemspezifische Sicherheitsprüfungen.

Nicht jeder Test kann auf jedem System ausgeführt werden.

Fehlen Programme, die ein bestimmter Test benötigt, kann Rkhunter diesen Test überspringen.

Grundlegender Systemcheck

Die zentrale Prüfung wird mit

```
sudo rkhunter --check
```

gestartet.

Die Kurzform lautet:

```
sudo rkhunter -c
```

Rkhunter führt damit seine aktivierten Prüfungen aus und zeigt die Ergebnisse im Terminal an.

Wird etwas als verdächtig eingestuft, erzeugt Rkhunter eine Warnung.

Zusätzlich wird standardmäßig ein Protokoll der durchgeführten Tests und Ergebnisse erzeugt.

Interaktiver Ablauf

Bei einem normalen `--check` kann Rkhunter nach bestimmten Testabschnitten darauf warten, dass der Benutzer mit der Eingabetaste fortfährt.

Für einen durchgehenden Scan steht deshalb

```
--skip-keypress
```

beziehungsweise

```
--sk
```

zur Verfügung.

Beispiel:

```
sudo rkhunter --check --skip-keypress
```

oder:

```
sudo rkhunter -c --sk
```

Damit läuft die Prüfung ohne diese manuellen Unterbrechungen bis zum Ende durch.

Warnungen

Eine Rkhunter-Warnung ist zunächst ein **technischer Hinweis auf einen prüfungsbedürftigen Zustand**.

Sie ist nicht automatisch der Beweis für ein Rootkit oder eine erfolgreiche Kompromittierung.

Das gilt insbesondere für Prüfungen auf veränderte Dateien.

Eine Datei kann sich aus legitimen Gründen verändern, beispielsweise durch ein reguläres Softwareupdate.

Rkhunter kann feststellen, dass bestimmte Eigenschaften von einem gespeicherten Referenzzustand abweichen. Daraus folgt jedoch nicht automatisch, warum diese Änderung erfolgt ist.

Kein Fund ist keine Sicherheitsgarantie

Umgekehrt bedeutet ein vollständig durchlaufener Rkhunter-Check ohne Warnung nicht, dass ein System garantiert kompromittierungsfrei ist.

Rkhunter kann nur nach den Merkmalen und Zuständen suchen, für die entsprechende Prüfverfahren implementiert sind.

Unbekannte, neue oder technisch anders arbeitende Angriffe können außerhalb dieser Erkennung liegen.

Rkhunter ist deshalb ein Diagnosewerkzeug und kein mathematischer Nachweis der Integrität eines Systems.

Unterstützte Prüfungen anzeigen

Mit

```
rkhunter --list tests
```

können die verfügbaren Testnamen angezeigt werden.

Damit lässt sich nachvollziehen, welche Tests von der installierten Version bereitgestellt werden.

Mit

```
rkhunter --list rootkits
```

kann die Liste der Rootkits angezeigt werden, nach denen Rkhunter sucht.

Weitere `--list`-Varianten können unter anderem Informationen über Sprachen, Perl-Unterstützung und die für die Dateieigenschaftsdatenbank verwendeten Dateien ausgeben.

Einzelne Prüfungen aktivieren

Mit

```
--enable
```

können gezielt bestimmte Tests aktiviert werden.

Grundprinzip:

```
rkhunter --check --enable TEST
```

Damit kann eine Untersuchung auf ausgewählte Testgruppen begrenzt werden.

Die tatsächlich verfügbaren Testnamen sollten zuvor mit

```
rkhunter --list tests
```

ermittelt werden.

Prüfungen deaktivieren

Mit

```
--disable
```

können bestimmte Tests deaktiviert werden.

Grundprinzip:

```
rkhunter --check --disable TEST
```

Auch hierbei sollten ausschließlich tatsächlich vorhandene Testnamen verwendet werden.

Das gezielte Abschalten einer Warnung sollte nicht dazu dienen, einen unbekannten Befund lediglich aus der Ausgabe zu entfernen.

Zunächst muss geklärt werden, warum die betreffende Prüfung eine Warnung erzeugt.

Konfiguration

Rkhunter verwendet eine zentrale Konfigurationsdatei:

```
rkhunter.conf
```

Zusätzlich kann das Programm eine lokale Konfigurationsdatei

```
rkhunter.conf.local
```

sowie ein Verzeichnis

```
rkhunter.d
```

mit zusätzlichen `.conf`-Dateien verwenden.

Kommandozeilenoptionen können entsprechende Einstellungen aus den Konfigurationsdateien überschreiben.

Die tatsächliche Konfiguration des installierten Systems ist für die Interpretation eines Scans relevant.

Konfiguration überprüfen

Mit

```
rkhunter --config-check
```

kann die Konfiguration überprüft werden.

Kurzform:

```
rkhunter -C
```

Rkhunter kontrolliert dabei die verwendeten Konfigurationsoptionen und meldet erkannte Konfigurationsprobleme.

Nach Änderungen an der Konfiguration ist eine solche Prüfung sinnvoll.

Dateieigenschaften

Eine wichtige Funktion von Rkhunter ist die Überwachung bestimmter Eigenschaften wichtiger Systemdateien und Kommandos.

Dafür speichert Rkhunter Referenzwerte in seiner Dateieigenschaftsdatenbank und vergleicht diese später mit dem aktuellen Zustand.

Dabei können unter anderem Hashwerte verwendet werden.

Standardmäßig verwendet Rkhunter für diese Funktion SHA-256, sofern die Konfiguration beziehungsweise Systemumgebung nichts anderes vorgibt.

Hashverfahren

Rkhunter unterstützt für die Dateieigenschaftsprüfung mehrere Hashverfahren.

Dazu gehören:

MD5,

SHA-1,

SHA-224,

SHA-256,

SHA-384,

SHA-512.

Über

```
--hash
```

kann das verwendete Verfahren beeinflusst werden.

Für die normale Rescue-Anwendung sollte die vorhandene Konfiguration jedoch nicht ohne technischen Grund verändert werden.

Properties-Datenbank

Die gespeicherten Dateieigenschaften bilden einen Referenzzustand.

Bei späteren Prüfungen kann Rkhunter feststellen, dass sich Eigenschaften einer überwachten Datei gegenüber diesem Referenzzustand verändert haben.

Diese Funktion ist nur dann aussagekräftig, wenn der gespeicherte Referenzzustand vertrauenswürdig ist.

Ein bereits kompromittierter Zustand darf nicht einfach zur neuen vertrauenswürdigen Referenz erklärt werden.

`--propupd`

Mit

```
rkhunter --propupd
```

wird die gespeicherte Dateieigenschaftsdatenbank aktualisiert.

Dieser Befehl ist sicherheitskritisch.

Rkhunter weist ausdrücklich darauf hin, dass das Programm lediglich erkennen kann, **dass** sich eine Datei verändert hat. Es kann anhand dieser Tatsache allein nicht feststellen, wodurch die Veränderung verursacht wurde.

Wird nach einer Veränderung

```
rkhunter --propupd
```

ausgeführt, übernimmt Rkhunter die aktuellen Eigenschaften als neuen Referenzzustand.

Damit wird implizit angenommen, dass die betreffende Datei vertrauenswürdig ist.

`--propupd` nicht blind verwenden

Aus diesem Grund darf

```
rkhunter --propupd
```

nicht einfach ausgeführt werden, um Warnungen nach einem Scan zu beseitigen.

Zuerst muss geklärt werden, warum sich die Datei verändert hat.

Beispiele für legitime Ursachen können reguläre Paketupdates sein.

Eine Veränderung kann aber ebenso Bestandteil einer Manipulation sein.

Erst wenn der neue Zustand als vertrauenswürdig bewertet wurde, darf er als neue Referenz übernommen werden.

Dies ist einer der wichtigsten Sicherheitshinweise bei der Arbeit mit Rkhunter.

Selektives `--propupd`

Rkhunter kann die Properties-Datenbank auch gezielt für einzelne Dateien, Verzeichnisse beziehungsweise Paketnamen aktualisieren.

Damit muss nicht zwangsläufig die gesamte Datenbank neu erzeugt werden.

Auch bei einer selektiven Aktualisierung gilt jedoch:

Nur ein verifizierter Zustand darf als neue Referenz gespeichert werden.

Paketmanager-Unterstützung

Rkhunter kann bei der Dateieigenschaftsprüfung Informationen eines unterstützten Paketmanagers verwenden.

Die Option

`--pkgmgr`

unterstützt unter anderem:

`DPKG`

und weitere Paketmanager.

Damit können Dateieigenschaften im Zusammenhang mit den vom Paketmanager bereitgestellten Informationen behandelt werden.

Der Einsatz hängt von der Konfiguration und dem untersuchten System ab.

Rkhunter-Daten aktualisieren

Rkhunter besitzt den Befehl:

```
rkhunter --update
```

Dieser Befehl aktualisiert **nicht das Rkhunter-Programm selbst**.

Er prüft vielmehr, ob neuere Versionen der von Rkhunter verwendeten Text-Datendateien verfügbar sind und lädt diese gegebenenfalls herunter.

Für diese Funktion wird eine geeignete Netzwerkverbindung und ein unterstütztes Kommandozeilen-Downloadwerkzeug benötigt.

Unterschied zwischen `--update` und Softwareupdate

Die Begriffe müssen sauber getrennt werden.

```
rkhunter --update
```

aktualisiert Rkhunter-Datendateien.

Es führt kein allgemeines Debian-/MX-Linux-Paketupdate durch und ersetzt nicht das installierte Paket `rkhunter`.

Die installierte Programmversion der Baseline bleibt:

Rkhunter 1.4.6 / Debian-Paket 1.4.6-13.

Versionsprüfung

Mit

```
rkhunter --versioncheck
```

kann geprüft werden, ob eine neuere Programmversion bekannt ist.

Auch diese Funktion benötigt Netzwerkzugriff.

Sie ist von

`rkhunter --update`

zu unterscheiden.

`--versioncheck` prüft die Programmversion.

`--update` prüft die Rkhunter-Datendateien.

Offline-Betrieb

JUMBO BLOCK Rescue kann ohne Internetverbindung betrieben werden.

In diesem Fall können die lokalen Rkhunter-Prüfungen grundsätzlich weiterhin mit den vorhandenen Daten und der vorhandenen Konfiguration ausgeführt werden.

Online-Funktionen wie

`--update`

oder

`--versioncheck`

können ohne erreichbare Updatequelle dagegen nicht erfolgreich durchgeführt werden.

Ein Offline-Check ist deshalb eine Untersuchung mit dem lokal vorhandenen Rkhunter-Stand.

Kein Netzwerkzwang

In isolierten Sicherheits-, KRITIS- oder forensischen Umgebungen darf eine Netzwerkverbindung nicht allein deshalb hergestellt werden, weil Rkhunter Daten aktualisieren könnte.

Die Sicherheitsvorgaben des Einsatzes haben Vorrang.

Ein vorhandener lokaler Rkhunter-Stand kann weiterhin für Untersuchungen verwendet werden, wobei dessen Aktualitätsstand bei der Bewertung berücksichtigt werden muss.

Protokollierung

Rkhunter erzeugt bei einem normalen Check standardmäßig eine Logdatei.

Der dokumentierte Standardpfad lautet:

`/var/log/rkhunter.log`

Mit

`--logfile`

beziehungsweise

`-l`

kann ein anderer Speicherort angegeben werden.

Das Protokoll enthält wesentlich mehr Informationen als lediglich die abschließende Zusammenfassung und sollte bei Warnungen ausgewertet werden.

Vorheriges Protokoll

Standardmäßig erzeugt Rkhunter bei einem neuen Lauf eine neue Logdatei.

Eine bereits vorhandene Logdatei wird dabei mit der Erweiterung

`.old`

umbenannt.

Mit

`--appendlog`

kann stattdessen an eine vorhandene Logdatei angehängt werden.

Für dokumentierte Rescue-Untersuchungen sollte bewusst entschieden werden, wie Scanprotokolle aufbewahrt werden.

Nur Warnungen anzeigen

Mit

`--report-warnings-only`

beziehungsweise

`--rwo`

kann die Terminalausgabe auf Warnmeldungen beschränkt werden.

Dies ist insbesondere bei automatisierten oder umfangreichen Prüfungen hilfreich.

Für eine erste detaillierte Analyse kann jedoch die vollständige Ausgabe beziehungsweise die Logdatei informativer sein.

Zusammenfassung

Nach einem normalen `--check` zeigt Rkhunter standardmäßig eine kurze Zusammenfassung der Ergebnisse.

Mit

`--nosummary`

beziehungsweise

`--ns`

kann diese unterdrückt werden.

Für einen manuellen Rescue-Einsatz ist die Zusammenfassung in der Regel hilfreich und sollte nicht ohne Grund abgeschaltet werden.

Farben

Rkhunter verwendet standardmäßig Farben für die Darstellung seiner Testergebnisse.

Erfolgreiche Prüfungen, Warnungen und übersprungene Tests können dadurch visuell unterschieden werden.

Mit

```
--nocolors
```

kann die Farbausgabe abgeschaltet werden.

Dies kann insbesondere für Protokollierung oder automatisierte Verarbeitung sinnvoll sein.

Quiet-Modus

Mit

```
--quiet
```

beziehungsweise

```
-q
```

kann die normale Ausgabe unterdrückt werden.

Dieser Modus ist vor allem dann sinnvoll, wenn lediglich Rückgabewerte oder gezielt aktivierte Ausgaben ausgewertet werden sollen.

Für eine interaktive Sicherheitsuntersuchung ist eine sichtbare Ausgabe meist hilfreicher.

Netzwerkprüfungen

Rkhunter führt verschiedene Prüfungen im Netzwerkbereich durch.

Die Programmdokumentation nennt ausdrücklich Untersuchungen von Netzwerkinterfaces und lauschenden Anwendungen.

Solche Prüfungen können Hinweise auf unerwartete Dienste oder Netzwerkzustände liefern.

Ein offener beziehungsweise lauschender Port ist jedoch nicht automatisch ein Rootkit-Beweis.

Er muss dem tatsächlich erwarteten Systemzustand gegenübergestellt werden.

Systemstartdateien

Rkhunter untersucht auch bestimmte Systemstartdateien auf Veränderungen beziehungsweise verdächtige Zustände.

Dies ist relevant, weil persistente Schadsoftware versuchen kann, sich über Startmechanismen dauerhaft in einem System zu verankern.

Auch hier gilt:

Eine Veränderung kann legitim sein und muss im Kontext bewertet werden.

Versteckte Dateien und Verzeichnisse

Rkhunter besitzt Prüfungen auf bestimmte versteckte beziehungsweise verdächtige Dateien und Verzeichnisse.

Unix-/Linux-Systeme verwenden jedoch regulär zahlreiche Dateien und Verzeichnisse, deren Namen mit einem Punkt beginnen.

„Versteckt“ bedeutet daher nicht automatisch „schädlich“.

Entscheidend ist, ob ein Fund für das betreffende System erwartet und erklärbar ist.

Rootkit-spezifische Prüfungen

Rkhunter enthält Tests für zahlreiche bekannte Rootkits und Malware-Merkmale.

Version 1.4.6 ergänzte unter anderem Prüfungen für weitere Malware beziehungsweise Backdoors sowie einen Test auf die Diamorphine-LKM-Technik.

Auch der Ebury-`sshd`-Backdoor-Komplex wurde in dieser Version durch zusätzliche Prüfungen berücksichtigt.

Diese Tests sind versionsspezifisch und Teil der Erkennungslogik von Rkhunter 1.4.6.

Shared-Memory-Prüfung

Rkhunter 1.4.6 enthält Prüfungen verdächtiger Shared-Memory-Segmente.

Die Version führte zusätzliche Konfigurationsmöglichkeiten ein, um bestimmte PIDs oder Benutzer bei dieser Prüfung gezielt freizugeben und eine Mindestgröße für zu prüfende Shared-Memory-Segmente festzulegen.

Auch hier können legitime Anwendungen auffällige Zustände erzeugen.

Eine Warnung muss deshalb untersucht werden.

Zusätzliche Hilfsprogramme

Einige Rkhunter-Prüfungen können zusätzliche externe Programme verwenden.

Sind benötigte Programme nicht installiert, können entsprechende Tests übersprungen werden.

Das Rkhunter-Projekt nennt beispielsweise `unhide` als optionales Werkzeug für zusätzliche Prüfungen.

Für die Interpretation eines Reports ist deshalb wichtig, nicht nur nach Warnungen zu schauen, sondern auch darauf, welche Tests tatsächlich ausgeführt beziehungsweise übersprungen wurden.

Chkrootkit und Rkhunter

Chkrootkit und Rkhunter verfolgen ein ähnliches Ziel, arbeiten jedoch nicht identisch.

Chkrootkit

führt spezialisierte Prüfungen auf bekannte Rootkit-Merkmale, manipulierte Programme, versteckte Prozesse und bestimmte Login- beziehungsweise Netzwerkauffälligkeiten durch.

Rkhunter

kombiniert Rootkit- und Malware-Prüfungen zusätzlich mit einer umfangreichen Überwachung bestimmter Systemdateieigenschaften, Systemkonfigurationen, Startdateien und weiterer Sicherheitszustände.

Die Programme ergänzen sich.

Ein Fund des einen Programms ist nicht automatisch durch das andere reproduzierbar.

Rkhunter und ClamAV

Auch Rkhunter ersetzt keinen ClamAV-Scan.

ClamAV untersucht zugängliche Dateien mit seiner Malware-Erkennungsengine.

Rkhunter untersucht den Zustand und bestimmte Eigenschaften eines Linux-/Unix-Systems.

Damit decken beide Werkzeuge unterschiedliche Sicherheitsaspekte ab.

Besonderheit eines Rescue-Systems

Rkhunter ist laut eigener Dokumentation primär für Prüfungen des **lokalen Systems** ausgelegt.

Das unterscheidet es in einem wichtigen Punkt von der in A.12 beschriebenen Chkrootkit-Funktion `-r`, mit der ausdrücklich ein alternatives Root-Verzeichnis angegeben werden kann.

Für Rkhunter 1.4.6 ist in der dokumentierten Kommandozeilenschnittstelle keine gleichwertige allgemeine `--rootdir`-Option vorhanden.

Deshalb darf nicht behauptet werden, Rkhunter könne ein beliebig eingehängtes fremdes Linux-System auf dieselbe Weise wie Chkrootkit mit `-r` vollständig als alternatives Root-System untersuchen.

Konsequenz für JUMBO BLOCK Rescue

Innerhalb von JUMBO BLOCK Rescue ist Rkhunter deshalb besonders geeignet zur Prüfung der **aktuell laufenden Linux-Umgebung beziehungsweise eines Systems, in dessen Systemkontext Rkhunter korrekt ausgeführt wird**.

Für die direkte Untersuchung eines lediglich unter einem Mountpunkt eingehängten fremden Linux-Dateisystems ist Chkrootkit mit seiner dokumentierten `-r`-Funktion flexibler.

Diese Einschränkung ist wichtig, damit Rkhunter im Handbuch keine nicht belegte Offline-Root-Funktion zugeschrieben wird.

Warnungen richtig behandeln

Bei einer Rkhunter-Warnung sollte zunächst festgestellt werden:

Welcher Test hat die Warnung erzeugt?

Welche Datei oder welcher Zustand ist betroffen?

Ist die Abweichung auf ein legitimes Paket- oder Systemupdate zurückzuführen?

Ist der aktuelle Zustand aus einer vertrauenswürdigen Quelle nachvollziehbar?

Gibt es weitere unabhängige Hinweise auf eine Kompromittierung?

Erst danach sollte entschieden werden, wie mit dem Befund umzugehen ist.

`--propupd` niemals als Warnungsbeseitigung

Besondere Vorsicht gilt nochmals für:

```
rkhunter --propupd
```

Dieser Befehl darf nicht als Mittel verstanden werden, Warnungen „zu reparieren“.

Er aktualisiert den Referenzzustand.

Wird er auf einem kompromittierten System ausgeführt, können manipulierte Dateien als neuer Normalzustand übernommen werden.

Deshalb gilt:

Erst verifizieren, dann Referenz aktualisieren.

Physisch beschädigte Datenträger

Rkhunter ist kein Datenrettungswerkzeug.

Bei einem physisch instabilen Laufwerk haben Datensicherung und Imaging Vorrang.

Ein umfangreicher Sicherheitscheck erzeugt zahlreiche Lesezugriffe und sollte nicht vor einer dringend notwendigen Sicherung eines ausfallgefährdeten Datenträgers erzwungen werden.

Für die eigentliche Rettung ist beispielsweise GNU ddrescue vorgesehen.

Keine automatische Bereinigung

Rkhunter ist ein Prüf- und Analysewerkzeug.

Eine Warnung führt nicht automatisch zu einer sicheren Bereinigung des Systems.

Bei einer bestätigten Kompromittierung muss separat entschieden werden, ob eine forensische Sicherung, Neuinstallation, Wiederherstellung aus einem vertrauenswürdigen Backup oder eine andere Reaktionsmaßnahme erforderlich ist.

Rkhunter trifft diese Entscheidung nicht.

Grenzen

Rkhunter erkennt nicht jede mögliche Schadsoftware oder jedes Rootkit.

Die Erkennung hängt von seinen implementierten Tests, Datendateien, der Konfiguration und den verfügbaren Hilfsprogrammen ab.

False Positives sind möglich.

Ein veränderter Hash beweist nicht die Ursache der Veränderung.

Ein unauffälliger Scan beweist keine vollständige Systemintegrität.

Und eine Warnung ist zunächst ein Indikator, der technisch untersucht werden muss.

Sicherheitshinweise

Rkhunter sollte für umfassende Systemprüfungen mit den erforderlichen Rechten ausgeführt werden.

Warnungen dürfen nicht ungeprüft als bestätigte Infektion interpretiert werden.

Ebenso dürfen sie nicht einfach durch Whitelisting oder `--propupd` beseitigt werden.

Vor `--propupd` muss der aktuelle Dateizustand als vertrauenswürdig verifiziert sein.

Online-Aktualisierungen dürfen in isolierten oder forensischen Umgebungen nicht entgegen den Sicherheitsvorgaben erzwungen werden.

Das Scanprotokoll sollte bei auffälligen Ergebnissen erhalten und ausgewertet werden.

Typischer Rescue-Ablauf

Für eine normale Prüfung des laufenden Systems kann zunächst die Konfiguration kontrolliert werden:

```
sudo rkhunter --config-check
```

Anschließend kann ein vollständiger Check ohne Unterbrechungen erfolgen:

```
sudo rkhunter --check --skip-keypress
```

Bei einer großen Ausgabe kann anschließend das Protokoll ausgewertet werden.

Für eine gezielte Anzeige von Warnungen steht:

```
sudo rkhunter --check --skip-keypress --report-warnings-only
```

zur Verfügung.

Werden Veränderungen wichtiger Dateien gemeldet, werden diese zunächst unabhängig verifiziert.

Erst wenn ihre Legitimität feststeht, darf eine Aktualisierung der Properties-Datenbank erwogen werden.

Rolle in JUMBO BLOCK Rescue

Rkhunter bildet innerhalb von JUMBO BLOCK Rescue eine zusätzliche Ebene der **hostbasierten Linux-/Unix-Sicherheitsdiagnose**.

Die Sicherheitswerkzeuge ergänzen sich:

ClamAV Scan – dateibasierte Malware-Untersuchung

ClamAV Signaturen – Versorgung der ClamAV-Engine mit Erkennungsdaten

Chkrootkit – Suche nach Rootkit- und Manipulationshinweisen, einschließlich der dokumentierten Untersuchung eines alternativen Root-Verzeichnisses

Rkhunter – Rootkit-, Malware-, Dateieigenschafts-, Startdatei- und weitere hostbasierte Sicherheitsprüfungen

GUFW Firewall – grafische Verwaltung der hostbasierten Firewall

Die besondere Stärke von Rkhunter liegt dabei in der Kombination aus bekannten Rootkit-/Malware-Prüfungen und der Überwachung sicherheitsrelevanter Systemeigenschaften.

Seine wichtigste methodische Grenze lautet gleichzeitig:

Rkhunter meldet Abweichungen und verdächtige Zustände – es beweist nicht allein deren Ursache.

Innerhalb von JUMBO BLOCK Rescue ist Rkhunter deshalb als Diagnose- und Kontrollinstrument zu verstehen, dessen Warnungen anschließend technisch bewertet werden müssen.

A.14 GUFW Firewall

Bereich: Sicherheit
Programm: GUFW Firewall
Baseline 20261001: GUFW 24.04.0
Installiertes Paket: `gufw 24.04.0-3`
Firewall-Grundlage: UFW 0.36.2
Installiertes Paket: `ufw 0.36.2-9`
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Konfiguration der hostbasierten Firewall von JUMBO BLOCK Rescue

Zweck

GUFW stellt eine grafische Benutzeroberfläche zur Verwaltung der hostbasierten Firewall bereit.

Der Name steht für:

Graphical Uncomplicated Firewall

GUFW ist dabei nicht selbst die grundlegende Firewall-Technik.

Es stellt eine grafische Oberfläche für **UFW – Uncomplicated Firewall** bereit.

In der JUMBO BLOCK Rescue Baseline 20261001 sind beide Komponenten installiert:

`gufw 24.04.0-3`

und

`ufw 0.36.2-9`.

Damit besteht die funktionale Kette aus:

GUFW → grafische Konfiguration

UFW → Verwaltung der Firewall-Regeln

Linux-Netzwerkfilterung → technische Durchsetzung der Regeln im Betriebssystem

Einsatzgebiet im Rescue-System

Ein Rescue-System wird häufig in Netzwerkumgebungen eingesetzt, deren Vertrauenswürdigkeit nicht vorausgesetzt werden kann.

Beispiele sind:

fremde Unternehmensnetze,

Notfallnetze,

temporäre Wartungsnetze,

direkte Ethernet-Verbindungen,

WLAN-Netze,

oder Netzwerke eines möglicherweise kompromittierten Standorts.

GUFW ermöglicht es, die lokale Firewall von JUMBO BLOCK Rescue grafisch zu kontrollieren und bei Bedarf Regeln für ein- und ausgehenden Netzwerkverkehr zu definieren.

Damit ist GUFW kein Werkzeug zur Malware-Erkennung.

Es kontrolliert Netzwerkzugriffe auf beziehungsweise vom laufenden Rescue-System entsprechend der konfigurierten Firewall-Regeln.

GUFW und UFW

GUFW und UFW müssen klar unterschieden werden.

GUFW

ist die grafische Benutzeroberfläche.

UFW

ist das darunterliegende Firewall-Konfigurationswerkzeug.

Änderungen, die über GUFW vorgenommen werden, wirken deshalb auf die von UFW verwaltete Firewall-Konfiguration.

GUFW stellt die wesentlichen Funktionen in einer grafisch zugänglichen Form bereit, sodass für typische Firewall-Aufgaben keine manuelle Eingabe von UFW-Kommandos erforderlich ist.

Firewall aktivieren und deaktivieren

GUFW besitzt einen zentralen Statusschalter für die Firewall.

Damit kann die Firewall aktiviert beziehungsweise deaktiviert werden.

Die entsprechende UFW-Funktion lautet auf Kommandozeilenebene grundsätzlich:

```
sudo ufw enable
```

beziehungsweise

```
sudo ufw disable
```

Das Deaktivieren der Firewall entfernt nicht zwangsläufig die gespeicherte Regeldefinition.

Es deaktiviert deren Anwendung durch UFW.

Für einen Rescue-Einsatz sollte die Firewall nicht ohne konkreten Grund deaktiviert werden.

Status prüfen

Auf Kommandozeilenebene kann der Status mit

```
sudo ufw status
```

überprüft werden.

Eine ausführlichere Darstellung liefert:

```
sudo ufw status verbose
```

Damit lässt sich unter anderem feststellen, ob UFW aktiv ist und welche Standardrichtlinien beziehungsweise Regeln wirksam sind.

GUFW stellt entsprechende Informationen grafisch dar.

Grundprinzip einer hostbasierten Firewall

Eine hostbasierte Firewall schützt den Rechner, auf dem sie ausgeführt wird.

Sie kann Netzwerkverkehr anhand definierter Regeln zulassen oder blockieren.

Dabei können unter anderem berücksichtigt werden:

Richtung des Verkehrs,

Protokoll,

Port,

IP-Adresse,

Subnetz,

und konkrete Regeldefinitionen.

GUFW beziehungsweise UFW ist damit von einer externen Netzwerkfirewall zu unterscheiden.

Es schützt unmittelbar das laufende JUMBO BLOCK Rescue System.

Eingehender und ausgehender Verkehr

GUFW unterscheidet zwischen:

Incoming – eingehendem Verkehr

und

Outgoing – ausgehendem Verkehr.

Eingehender Verkehr umfasst Verbindungsversuche beziehungsweise Datenverkehr, der das Rescue-System von außen erreicht.

Ausgehender Verkehr wird vom Rescue-System in Richtung anderer Systeme beziehungsweise Netze initiiert.

Für beide Richtungen können Richtlinien festgelegt werden.

Standardrichtlinien

GUFW ermöglicht die Auswahl von Standardrichtlinien für eingehenden und ausgehenden Verkehr.

UFW arbeitet ebenfalls mit solchen Default Policies.

Das Grundprinzip lautet:

Wenn keine speziellere Regel auf einen Datenverkehr zutrifft, entscheidet die jeweilige Standardrichtlinie.

Damit bilden Standardrichtlinien die Basis der Firewall-Konfiguration.

Typischer Sicherheitsansatz

Ein verbreiteter Host-Firewall-Ansatz besteht darin:

eingehende Verbindungen standardmäßig zu blockieren

und

ausgehende Verbindungen standardmäßig zu erlauben.

Das ist jedoch keine universell vorgeschriebene Konfiguration.

In besonders restriktiven oder isolierten Einsatzumgebungen kann auch ausgehender Verkehr begrenzt werden.

Die tatsächlich erforderliche Firewall-Policy hängt vom Rescue-Szenario und den Sicherheitsvorgaben des jeweiligen Netzes ab.

Regeln

GUFW ermöglicht das Anlegen eigener Firewall-Regeln.

Eine Regel kann Netzwerkverkehr beispielsweise:

Allow – erlauben

Deny – ablehnen beziehungsweise blockieren

Reject – zurückweisen

Limit – begrenzen

je nach verfügbarer Regelart und Konfiguration.

Die genaue Wirkung muss entsprechend der gewählten UFW-Regel verstanden werden.

Allow

Eine Allow-Regel erlaubt den passenden Netzwerkverkehr.

Auf UFW-Ebene lautet ein einfaches Beispiel:

```
sudo ufw allow 22
```

Damit würde Verkehr entsprechend der UFW-Regel für Port 22 erlaubt.

Eine solche Regel sollte nur eingerichtet werden, wenn der betreffende Dienst tatsächlich benötigt wird.

Deny

Eine Deny-Regel blockiert passenden Netzwerkverkehr.

Beispielprinzip:

```
sudo ufw deny 22
```

Die Gegenstelle erhält bei einem blockierten Paket nicht zwangsläufig dieselbe unmittelbare Rückmeldung wie bei einer Reject-Regel.

Reject

Eine Reject-Regel verweigert den Verkehr und signalisiert der Gegenstelle die Ablehnung.

Damit unterscheidet sich `reject` in seinem Netzwerkverhalten von einem einfachen `deny`.

Welche Variante zweckmäßig ist, hängt vom Einsatzfall ab.

Limit

UFW besitzt eine Limit-Funktion.

Sie kann insbesondere verwendet werden, um häufige neue Verbindungsversuche auf einen Dienst zu begrenzen.

Die UFW-Dokumentation nennt SSH als typisches Beispiel.

Grundprinzip:

```
sudo ufw limit ssh
```

UFW begrenzt damit neue Verbindungsversuche anhand seiner implementierten Rate-Limit-Logik.

Dies kann einen Dienst gegen einfache wiederholte Verbindungsversuche zusätzlich absichern.

Es ersetzt jedoch keine vollständige Authentifizierungs- oder Angriffsschutzlösung.

Ports

Firewall-Regeln können sich auf bestimmte Netzwerkports beziehen.

Beispiele typischer Portnummern sind:

22 für SSH,

53 für DNS,

80 für HTTP,

443 für HTTPS.

Eine Portnummer allein sagt jedoch nicht mit Sicherheit aus, welcher konkrete Prozess tatsächlich darauf lauscht.

Vor dem Freigeben eines Ports sollte daher bekannt sein, welcher Dienst ihn verwendet und warum die Freigabe benötigt wird.

Protokolle

Regeln können nach Netzwerkprotokollen differenziert werden.

Insbesondere sind dabei:

TCP

und

UDP

relevant.

Ein Dienst kann TCP, UDP oder beide Protokolle verwenden.

Eine Firewall-Regel sollte deshalb zum tatsächlichen Netzwerkprotokoll des Dienstes passen.

IP-Adressen

UFW-Regeln können auf bestimmte Quell- oder Zieladressen eingeschränkt werden.

Damit muss ein Port nicht zwangsläufig für jedes erreichbare System freigegeben werden.

Ein Dienst kann beispielsweise ausschließlich für eine bestimmte administrative Gegenstelle oder ein bestimmtes Netz zugelassen werden.

Für Wartungs- und Rescue-Szenarien kann eine solche Einschränkung wesentlich sicherer sein als eine allgemeine Freigabe.

Subnetze

Regeln können ebenfalls auf IP-Netze beziehungsweise Subnetze angewendet werden.

Damit lässt sich beispielsweise festlegen, dass ein Dienst nur innerhalb eines bestimmten Wartungsnetzes erreichbar sein soll.

Die verwendeten Netzbereiche müssen dabei der tatsächlichen Infrastruktur entsprechen.

Sie dürfen im Handbuch nicht pauschal vorausgesetzt werden.

Regelreihenfolge

Firewall-Regeln besitzen eine Reihenfolge.

Diese Reihenfolge kann relevant sein, wenn mehrere Regeln denselben oder überlappenden Netzwerkverkehr betreffen.

Auf UFW-Kommandozeilenebene können Regeln deshalb auch gezielt an bestimmten Positionen eingefügt werden.

Bei komplexeren Konfigurationen sollte die Regelbasis als Ganzes geprüft werden und nicht lediglich eine einzelne Regel isoliert betrachtet werden.

Regeln löschen

Nicht mehr benötigte Regeln können entfernt werden.

Auf UFW-Ebene kann dies beispielsweise anhand einer Regeldefinition oder nummerierten Regelliste erfolgen.

Mit:

```
sudo ufw status numbered
```

können Regeln nummeriert angezeigt werden.

Anschließend kann eine Regel gezielt gelöscht werden.

Vor dem Löschen sollte geprüft werden, ob sie für eine bestehende Netzwerkfunktion benötigt wird.

Profile

GUFW unterstützt Profile.

Damit können unterschiedliche Firewall-Konfigurationen für verschiedene Einsatzsituationen verwaltet werden.

Das ist insbesondere für mobile Systeme sinnvoll, die in unterschiedlichen Netzwerkumgebungen eingesetzt werden.

Eine restriktive Konfiguration für ein unbekanntes Netzwerk kann sich beispielsweise von einer gezielt freigeschalteten Wartungsumgebung unterscheiden.

Die tatsächlich verwendeten Profile sind jedoch installationsabhängig.

Aus der Baseline-Paketliste allein lässt sich nicht ableiten, welche individuellen GUFW-Profile in JUMBO BLOCK Rescue vorkonfiguriert sind.

Logging

UFW unterstützt die Protokollierung von Firewall-Ereignissen.

Die Protokollierung kann unter anderem mit:

```
sudo ufw logging on
```

aktiviert werden.

UFW unterstützt unterschiedliche Logging-Level.

Firewall-Logging kann bei der Netzwerkdiagnose und Sicherheitsanalyse hilfreich sein.

Es kann jedoch bei hohem Netzwerkverkehr erhebliche Mengen an Protokolldaten erzeugen.

Der Logging-Level sollte deshalb dem tatsächlichen Analysebedarf entsprechen.

GUFW und Logging

GUFW stellt die UFW-Protokollierungsfunktion grafisch zur Verfügung.

Damit kann das Firewall-Logging ohne direkte Kommandozeilenkonfiguration gesteuert werden.

Die erzeugten Logs sind insbesondere dann nützlich, wenn nachvollzogen werden soll, ob Netzwerkverkehr aufgrund einer Firewall-Regel blockiert wurde.

Firewall und Netzwerkdiagnose

Eine aktive Firewall kann die Ergebnisse anderer Netzwerkwerkzeuge beeinflussen.

Das betrifft beispielsweise:

Nmap,

MTR,

Ping-basierte Diagnose,

bestimmte Wireshark-Szenarien,

SSH,

und andere Netzwerkdienste.

Wird eine erwartete Netzwerkverbindung nicht aufgebaut, muss deshalb auch geprüft werden, ob die lokale Firewall den betreffenden Verkehr blockiert.

Die Firewall sollte jedoch nicht vorschnell vollständig deaktiviert werden.

Zunächst sollte die konkrete Regel beziehungsweise der benötigte Netzwerkverkehr identifiziert werden.

GUFW und SSH

JUMBO BLOCK Rescue enthält OpenSSH.

Soll ein SSH-Dienst auf dem Rescue-System erreichbar gemacht werden, muss neben dem Dienst selbst auch die Firewall-Konfiguration berücksichtigt werden.

Eine Firewall-Freigabe startet jedoch keinen SSH-Server.

Ebenso bedeutet ein laufender SSH-Dienst nicht automatisch, dass er durch die Firewall erreichbar ist.

Dienststatus und Firewall-Regel sind zwei unterschiedliche Ebenen.

GUFW und Nmap

Nmap untersucht erreichbare Netzwerkdienste und Ports.

Die lokale Firewall des Rescue-Systems kann sowohl die Erreichbarkeit eigener Dienste als auch bestimmte Scan- beziehungsweise Antwortsituationen beeinflussen.

Bei der Interpretation von Nmap-Ergebnissen muss deshalb berücksichtigt werden, ob GUFW/UFW aktiv ist und welche Regeln gelten.

GUFW und Wireshark

Wireshark zeichnet Netzwerkverkehr auf, der an den überwachten Interfaces sichtbar ist.

GUFW beziehungsweise UFW entscheidet dagegen anhand der Firewall-Regeln über Netzwerkverkehr des Hosts.

Beide Programme erfüllen damit unterschiedliche Aufgaben:

GUFW/UFW – kontrolliert Verkehr

Wireshark – analysiert beziehungsweise zeichnet Verkehr auf

Wireshark ersetzt keine Firewall und GUFW ersetzt keinen Paketanalysator.

GUFW und ClamAV

Auch zwischen Firewall und Malware-Scanner besteht eine klare Aufgabentrennung.

GUFW untersucht keine Dateien auf Schadsoftware.

ClamAV kontrolliert keine Netzwerkports als Firewall.

Die Sicherheitsfunktionen ergänzen sich:

ClamAV → Malware-Erkennung

Chkrootkit/Rkhunter → Rootkit- und Systemprüfung

GUFW/UFW → Netzwerkzugriffskontrolle

Reset

UFW unterstützt:

```
sudo ufw reset
```

Dieser Befehl setzt die Firewall-Konfiguration auf den Installationsstandard zurück.

Dabei werden vorhandene benutzerdefinierte Regeln deaktiviert und zurückgesetzt.

Diese Funktion ist deshalb sicherheitskritisch.

Sie darf nicht als allgemeiner Problemlösungsschritt verwendet werden, wenn lediglich eine einzelne Verbindung nicht funktioniert.

Vor einem Reset muss bekannt sein, welche Regeln dadurch verloren gehen beziehungsweise verändert werden.

Firewall deaktivieren ist keine Diagnose

Wenn ein Netzwerkdienst nicht erreichbar ist, ist das vollständige Abschalten der Firewall zwar ein möglicher technischer Test, aber kein geeigneter Standardablauf.

Besser ist zunächst:

Firewall-Status prüfen,

Regelliste prüfen,

benötigten Port und das Protokoll bestimmen,

Quelle beziehungsweise Ziel feststellen,

und anschließend die konkrete Regel korrigieren.

So bleibt die Schutzwirkung für andere Netzwerkverbindungen erhalten.

Firewall schützt nicht vor allen Angriffen

Eine hostbasierte Firewall ist nur eine Sicherheitsebene.

Sie kann unerwünschten Netzwerkverkehr entsprechend ihren Regeln blockieren.

Sie verhindert jedoch beispielsweise nicht automatisch:

Schadsoftware in bereits vorhandenen Dateien,

Angriffe über ausdrücklich erlaubte Dienste,

Schwachstellen innerhalb freigegebener Anwendungen,

Manipulationen über physische Datenträger,

oder Benutzerfehler.

GUFW ist deshalb Bestandteil eines mehrschichtigen Sicherheitskonzepts und kein vollständiges Sicherheitssystem.

Ausgehender Verkehr

Bei einem Rescue-System verdient auch ausgehender Verkehr Beachtung.

Wird beispielsweise ein potenziell kompromittierter Datenträger untersucht, bedeutet dies nicht automatisch, dass darauf befindliche Schadsoftware ausgeführt wird.

Werden jedoch unbekannte Programme bewusst gestartet, können diese grundsätzlich Netzwerkkommunikation initiieren.

Eine restriktivere Outgoing-Policy kann in speziellen Analyseumgebungen eine zusätzliche Sicherheitsmaßnahme darstellen.

Sie kann gleichzeitig legitime Funktionen wie:

DNS,

Webzugriff,

FreshClam-Updates,

SSH,

Paketdownloads

oder andere Netzwerkdienste blockieren.

Eine solche Policy muss deshalb bewusst geplant werden.

Unbekannte Netzwerke

Wird JUMBO BLOCK Rescue mit einem unbekannten oder nicht vertrauenswürdigen Netzwerk verbunden, sollte die lokale Angriffsfläche möglichst klein gehalten werden.

Nicht benötigte eingehende Dienste sollten nicht unnötig freigegeben werden.

GUFW bietet dafür eine grafisch kontrollierbare Möglichkeit, die hostbasierte Firewall zu verwalten.

Rescue-System als Wartungsrechner

In bestimmten Szenarien muss JUMBO BLOCK Rescue aktiv Netzwerkdienste erreichen oder selbst bereitstellen.

Beispiele können sein:

SSH-Verbindungen,

DNS-Abfragen,

Webzugriffe,

Signaturupdates,

Netzwerkdiagnosen

oder Dateiübertragungen.

Firewall-Regeln müssen diese Anforderungen berücksichtigen.

Die Regel sollte dabei möglichst genau auf den tatsächlich benötigten Dienst zugeschnitten werden.

Firewall-Regeln vor Änderungen dokumentieren

Bei komplexeren Einsatzsituationen sollte vor umfangreichen Änderungen der aktuelle Zustand dokumentiert werden.

Auf UFW-Ebene kann beispielsweise:

```
sudo ufw status verbose
```

beziehungsweise:

```
sudo ufw status numbered
```

zur Bestandsaufnahme verwendet werden.

Dadurch lässt sich nachvollziehen, welche Regeln vor einer Änderung vorhanden waren.

Typischer Rescue-Ablauf

Vor der Verbindung mit einem unbekannten Netzwerk kann zunächst geprüft werden, ob die Firewall aktiv ist.

Grafisch erfolgt dies über GUFW.

Alternativ kann der Status kontrolliert werden mit:

```
sudo ufw status verbose
```

Anschließend wird geprüft, ob die Standardrichtlinien zum vorgesehenen Einsatz passen.

Werden zusätzliche Dienste benötigt, sollten nur die tatsächlich notwendigen Regeln angelegt werden.

Nach Änderungen wird der Status erneut geprüft.

Bei Netzwerkproblemen wird zuerst die konkrete Regelbasis untersucht, bevor die Firewall vollständig deaktiviert wird.

Sicherheitshinweise

GUFW verändert die aktive Firewall-Konfiguration des Rescue-Systems.

Eine falsche Regel kann notwendige Netzwerkkommunikation blockieren.

Eine zu weit gefasste Allow-Regel kann dagegen unnötige Netzwerkzugriffe ermöglichen.

Ports sollten nur freigegeben werden, wenn der zugehörige Dienst bekannt und erforderlich ist.

Wenn möglich, sollten Freigaben auf die tatsächlich benötigten Quelladressen beziehungsweise Netze beschränkt werden.

`ufw reset` löscht beziehungsweise setzt die bestehende benutzerdefinierte Regelkonfiguration zurück und darf nicht unüberlegt eingesetzt werden.

Eine deaktivierte Firewall bietet keine durch UFW verwaltete Filterwirkung.

Firewall-Regeln ersetzen keine Malware-Prüfung, Systemhärtung oder sichere Dienstkonfiguration.

Grenzen

GUFW ist eine grafische Firewall-Verwaltung und kein Intrusion-Detection-System.

Es erkennt keine Rootkits.

Es scannt keine Dateien auf Malware.

Es analysiert keine Netzwerkpakete wie Wireshark.

Es führt keine Portscans wie Nmap durch.

Es beseitigt keine Schwachstellen in Netzwerkdiensten.

Seine Aufgabe besteht darin, die hostbasierte Firewall-Konfiguration kontrollierbar zu machen.

Technische Einordnung in JUMBO BLOCK Rescue

GUFW bildet innerhalb von JUMBO BLOCK Rescue die grafische Ebene der **lokalen Netzwerkzugriffskontrolle**.

Die Sicherheitswerkzeuge des Rescue-Systems decken damit unterschiedliche Aufgaben ab:

A.10 ClamAV Scan
Untersuchung von Dateien und Verzeichnissen auf durch ClamAV erkennbare Malware.

A.11 ClamAV Signaturen
Pflege der ClamAV-Erkennungsdatenbanken.

A.12 Chkrootkit
Suche nach bestimmten Rootkit- und Manipulationshinweisen auf Unix-/Linux-Systemen.

A.13 Rkhunter
Zusätzliche Rootkit-, Malware-, Dateieigenschafts- und hostbasierte Sicherheitsprüfungen.

A.14 GUFW Firewall
Grafische Verwaltung der hostbasierten Netzwerkfirewall.

Damit schließt GUFW den Bereich Sicherheit nicht als weiteres Erkennungswerkzeug ab, sondern als **präventive und kontrollierende Netzwerkkomponente**.

Die klare technische Zuordnung lautet:

GUFW = grafische Bedienoberfläche

UFW = Firewall-Konfigurationsschicht

Firewall-Regeln = Kontrolle des ein- und ausgehenden Netzwerkverkehrs des Rescue-Systems

Für JUMBO BLOCK Rescue bedeutet dies insbesondere, dass das System auch bei Einsätzen in unbekannten oder nicht vertrauenswürdigen Netzwerken über eine unmittelbar bedienbare lokale Firewall-Konfiguration verfügt.

A.15 Binwalk

Bereich: Analyse
Programm: Binwalk
Baseline 20261001: Binwalk 2.4.3
Installiertes Paket: `binwalk 2.4.3+dfsg1-2+deb13u1`
Bedienung: Kommandozeile
Aufgabe: Analyse binärer Dateien und Images auf eingebettete Dateien, Datenstrukturen und ausführbaren Code

Zweck

Binwalk ist ein Werkzeug zur Analyse binärer Dateien und Images.

Sein primäres Einsatzgebiet ist die Analyse von Firmware-Images. Das Werkzeug kann jedoch allgemein Binärdaten nach bekannten Dateisignaturen und eingebetteten Datenstrukturen durchsuchen.

Binwalk kann dabei unter anderem:

bekannte Dateisignaturen erkennen,

Offsets erkannter Strukturen bestimmen,

eingebettete Inhalte extrahieren,

extrahierte Inhalte rekursiv weiter untersuchen,

nach bestimmten Bytefolgen suchen,

nach typischen ausführbaren Opcodes suchen,

und die Entropie von Binärdaten analysieren.

Damit unterscheidet sich Binwalk deutlich von den bisher dokumentierten Datenrettungs- und Sicherheitsprogrammen.

Binwalk beantwortet primär die Frage:

„Welche erkennbaren Datenstrukturen befinden sich innerhalb dieser Binärdatei und an welcher Position beginnen sie?“

Typische Einsatzgebiete

Binwalk wurde insbesondere für die Analyse und das Reverse Engineering von Firmware entwickelt.

Typische Untersuchungsobjekte sind beispielsweise:

Firmware-Images,

Binärabbilder,

Speicherabbilder,

zusammengesetzte Binärdateien,

Dateien mit eingebetteten Archiven,

komprimierte Datenbereiche,

Dateisystem-Images

und andere Binärcontainer.

Innerhalb von JUMBO BLOCK Rescue dient Binwalk damit als spezialisiertes Analysewerkzeug für Dateien und Images, deren interner Aufbau untersucht werden soll.

Grundlegender Aufruf

Die grundlegende Syntax lautet:

```
binwalk [OPTIONEN] DATEI
```

Ein einfacher Aufruf ist beispielsweise:

```
binwalk firmware.bin
```

Binwalk durchsucht die angegebene Datei nach bekannten Signaturen.

Für erkannte Strukturen werden unter anderem deren Positionen innerhalb der Datei ausgegeben.

Offsets

Ein wesentliches Ergebnis eines Binwalk-Scans ist der **Offset**.

Der Offset bezeichnet die Position innerhalb der untersuchten Datei, an der eine erkannte Struktur beziehungsweise Signatur beginnt.

Binwalk zeigt solche Positionen typischerweise sowohl dezimal als auch hexadezimal an.

Damit kann anschließend beispielsweise mit einem Hexeditor gezielt an dieser Position weiter untersucht werden.

Genau hier ergänzen sich:

Binwalk – Strukturen und Positionen finden

und

Hexedit – Binärdaten unmittelbar untersuchen und bearbeiten.

Signaturanalyse

Die zentrale Analysefunktion von Binwalk ist der Signatur-Scan.

Explizit kann dieser mit

```
-B
```

beziehungsweise

`--signature`

angefordert werden.

Beispiel:

```
binwalk -B firmware.bin
```

Binwalk durchsucht die Zieldatei dabei nach bekannten Dateisignaturen.

Solche Signaturen basieren auf charakteristischen Bytefolgen beziehungsweise Magic-Werten und zusätzlichen Erkennungsregeln.

Signaturfund ist kein vollständiger Beweis

Ein erkannter Magic-Wert beziehungsweise eine Signatur bedeutet zunächst, dass die untersuchten Bytes zu einer bekannten Struktur passen.

Das ist nicht in jedem Fall gleichbedeutend mit einer vollständig validierten oder intakten eingebetteten Datei.

Binärdaten können zufällig Bytefolgen enthalten, die bekannten Signaturen entsprechen.

Beschädigte oder abgeschnittene Daten können ebenfalls nur teilweise zu einer Signatur passen.

Die Ergebnisse müssen deshalb im Kontext der untersuchten Datei bewertet werden.

Ausgabe ungültiger Treffer

Mit

`-I`

beziehungsweise

`--invalid`

können auch Ergebnisse angezeigt werden, die von Binwalk als ungültig markiert wurden.

Diese Option ist vor allem für detaillierte Analysen interessant.

Für eine normale erste Untersuchung sind solche Treffer häufig weniger relevant.

Bestimmte Signaturen einschließen

Mit

`-y`

beziehungsweise

`--include`

kann die Ausgabe auf Signaturen beschränkt werden, deren Beschreibung beziehungsweise Kennzeichnung dem angegebenen Ausdruck entspricht.

Grundprinzip:

```
binwalk -y AUSDRUCK DATEI
```

Damit kann eine umfangreiche Analyse gezielt auf bestimmte Strukturen eingeschränkt werden.

Signaturen ausschließen

Das Gegenstück ist:

```
-x
```

beziehungsweise

```
--exclude
```

Damit können passende Signaturergebnisse aus der Analyse ausgeschlossen werden.

Grundprinzip:

```
binwalk -x AUSDRUCK DATEI
```

Dies kann bei Dateien mit sehr vielen bekannten, aber für die konkrete Untersuchung irrelevanten Strukturen hilfreich sein.

Suche nach einer Bytefolge

Mit

```
-R
```

beziehungsweise

```
--raw
```

kann Binwalk nach einer vorgegebenen Bytefolge suchen.

Grundprinzip:

```
binwalk -R ZEICHENFOLGE DATEI
```

Diese Funktion ist nützlich, wenn nicht nach einem von Binwalk bereits definierten Dateityp, sondern nach einer konkreten bekannten Bytefolge gesucht werden soll.

Opcode-Scan

Mit

```
-A
```

beziehungsweise

```
--opcodes
```

sucht Binwalk nach bekannten Signaturen ausführbarer Opcodes.

Beispiel:

```
binwalk -A firmware.bin
```

Diese Funktion kann Hinweise auf ausführbaren Maschinencode innerhalb eines Binärabbilds liefern.

Ein Opcode-Treffer allein bestimmt jedoch nicht automatisch die vollständige Architektur oder Funktion des gefundenen Codes.

Er ist ein Analysehinweis.

Eigene Magic-Datei

Mit

`-m`

beziehungsweise

`--magic`

kann eine eigene Magic-Datei angegeben werden.

Grundprinzip:

```
binwalk -m DATEI firmware.bin
```

Damit kann die Signaturanalyse um eigene beziehungsweise speziell angepasste Erkennungsdefinitionen erweitert werden.

Für normale Rescue-Untersuchungen ist dies nicht erforderlich.

Es ist eine Funktion für spezialisierte Binäranalysen.

Automatische Extraktion

Eine der wichtigsten Funktionen von Binwalk ist:

`-e`

beziehungsweise

`--extract`.

Beispiel:

```
binwalk -e firmware.bin
```

Binwalk versucht dabei, erkannte bekannte Dateitypen automatisch zu extrahieren.

Damit geht die Funktion über eine reine Signaturanalyse hinaus.

Der normale Signatur-Scan sagt:

„Hier befindet sich wahrscheinlich eine bekannte Struktur.“

Die Extraktionsfunktion versucht zusätzlich:

„Extrahiere diese erkannte Struktur.“

Externe Extraktionsprogramme

Bei der automatischen Extraktion kann Binwalk für bestimmte Dateitypen externe Hilfsprogramme verwenden.

Das bedeutet:

Ein Signaturtyp kann erkannt werden, obwohl das zur eigentlichen Extraktion benötigte externe Programm nicht verfügbar ist.

Erkennung und erfolgreiche Extraktion sind daher zwei unterschiedliche Fähigkeiten.

Ein erkannter Inhalt garantiert nicht, dass Binwalk ihn auf dem konkreten System automatisch entpacken beziehungsweise verarbeiten kann.

Extraktionsverzeichnis

Mit

`-C`

beziehungsweise

`--directory`

kann ein eigenes Zielverzeichnis für extrahierte Inhalte festgelegt werden.

Grundprinzip:

```
binwalk -e -C ZIELVERZEICHNIS firmware.bin
```

Dies ist insbesondere bei Rescue- und Analysearbeiten sinnvoll, damit erzeugte Dateien bewusst von der Quelldatei getrennt abgelegt werden.

Rekursive Extraktion

Binwalk unterstützt die sogenannte Matryoshka-Analyse.

Die Option lautet:

`-M`

beziehungsweise

`--matryoshka`.

Sie bewirkt, dass extrahierte Dateien erneut mit Binwalk untersucht werden.

In Kombination mit der Extraktion kann damit eine verschachtelte Struktur schrittweise analysiert werden.

Beispielprinzip:

```
binwalk -e -M firmware.bin
```

Eine Firmware kann beispielsweise ein komprimiertes Archiv enthalten, das wiederum ein Dateisystem oder weitere komprimierte Dateien enthält.

Die rekursive Analyse kann solche Ebenen weiterverfolgen.

Rekursionstiefe

Mit

`-d`

beziehungsweise

`--depth`

wird die maximale Rekursionstiefe der Matryoshka-Analyse begrenzt.

Der dokumentierte Standardwert der 2.x-Kommandozeilenschnittstelle beträgt:

8 Ebenen.

Beispiel:

```
binwalk -e -M -d 4 firmware.bin
```

Damit wird die rekursive Untersuchung auf vier Ebenen begrenzt.

Eine Begrenzung kann sinnvoll sein, um eine unerwartet umfangreiche oder zyklische Analyse zu vermeiden.

Anzahl extrahierter Dateien begrenzen

Mit

`-n`

beziehungsweise

`--count`

kann die Anzahl extrahierter Dateien begrenzt werden.

Dies kann bei unbekannten beziehungsweise sehr komplexen Images helfen, die Menge erzeugter Analysedaten zu kontrollieren.

Größe extrahierter Dateien begrenzen

Mit

`-j`

beziehungsweise

`--size`

kann die Größe einzelner extrahierter Dateien begrenzt werden.

Auch diese Funktion dient dazu, die automatische Extraktion kontrollierbarer zu machen.

Carving ohne Extraktionsprogramme

Mit

`-z`

beziehungsweise

`--carve`

kann Binwalk Daten aus Dateien herausschneiden, ohne die zugehörigen externen Extraktionsprogramme auszuführen.

Damit lässt sich zwischen:

Carving

und

weitergehender automatischer Extraktion beziehungsweise Verarbeitung

unterscheiden.

Diese Trennung kann bei unbekannten Untersuchungsobjekten sicherheitsrelevant sein.

Rekursive Analyse mit Vorsicht

Eine rekursive automatische Extraktion kann sehr viele Dateien erzeugen.

Sie kann außerdem externe Extraktionsprogramme auf erkannte Inhalte anwenden.

Bei unbekannten oder nicht vertrauenswürdigen Binärdaten sollte deshalb bewusst entschieden werden, ob lediglich eine Signaturanalyse oder tatsächlich eine automatische rekursive Extraktion erforderlich ist.

Für eine erste Untersuchung ist ein einfacher:

```
binwalk DATEI
```

deutlich zurückhaltender als eine sofortige rekursive Extraktion.

Entropieanalyse

Mit

`-E`

beziehungsweise

`--entropy`

berechnet Binwalk die Entropie der untersuchten Datei.

Beispiel:

```
binwalk -E firmware.bin
```

Die Entropieanalyse untersucht vereinfacht ausgedrückt, wie stark die Bytewerte in einzelnen Bereichen variieren beziehungsweise wie hoch deren statistische Unordnung ist.

Bedeutung hoher Entropie

Bereiche mit hoher Entropie können beispielsweise auf:

komprimierte Daten

oder

verschlüsselte Daten

hindeuten.

Das Binwalk-Projekt nennt die Entropieanalyse ausdrücklich als Möglichkeit, unbekannte komprimierte oder verschlüsselte Bereiche zu erkennen.

Wichtig ist jedoch:

Hohe Entropie beweist keine Verschlüsselung.

Komprimierte Daten können ebenfalls sehr hohe Entropiewerte besitzen.

Auch andere Daten können statistisch ähnlich erscheinen.

Die Entropieanalyse liefert daher Hinweise auf Datenstrukturen, aber keine automatische inhaltliche Klassifizierung.

Schnelle Entropieanalyse

Mit

`-F`

beziehungsweise

`--fast`

kann eine schnellere, dafür weniger detaillierte Entropieanalyse verwendet werden.

Dies kann bei großen Images für eine erste Übersicht sinnvoll sein.

Entropiediagramm speichern

Mit

`-J`

beziehungsweise

`--save`

kann das erzeugte Entropiediagramm als PNG-Datei gespeichert werden.

Dies ermöglicht eine visuelle Untersuchung des Entropieverlaufs entlang des analysierten Binärabbilds.

Scanbereich begrenzen

Binwalk muss nicht zwangsläufig eine gesamte Datei untersuchen.

Mit

`-o`

beziehungsweise

`--offset`

kann ein Startoffset angegeben werden.

Mit

`-l`

beziehungsweise

`--length`

kann die Anzahl der zu untersuchenden Bytes begrenzt werden.

Damit lässt sich eine Analyse gezielt auf einen bereits bekannten interessanten Bereich konzentrieren.

Basisadresse

Mit

`-O`

beziehungsweise

`--base`

kann ein Basiswert zu den ausgegebenen Offsets addiert werden.

Dies kann bei Analysen relevant sein, wenn die Datei einen Ausschnitt aus einem größeren Adressraum repräsentiert.

Für normale Signaturanalysen ist diese Funktion nicht erforderlich.

Byte-Reihenfolge verändern

Mit

`-g`

beziehungsweise

`--swap`

kann Binwalk Gruppen von Bytes vor der Analyse umkehren.

Diese Funktion ist für spezielle Binärformate beziehungsweise bestimmte Byte-Anordnungen vorgesehen.

Sie sollte nur verwendet werden, wenn der Aufbau des untersuchten Datenformats dies erfordert.

Ergebnisse protokollieren

Mit

`-f`

beziehungsweise

`--log`

können Analyseergebnisse in eine Datei geschrieben werden.

Grundprinzip:

```
binwalk -f ergebnis.txt firmware.bin
```

Damit lässt sich ein Scan für spätere Auswertung dokumentieren.

CSV-Ausgabe

Mit

`-c`

beziehungsweise

`--csv`

kann die Logausgabe im CSV-Format erfolgen.

Das ist insbesondere für eine strukturierte Weiterverarbeitung von Analyseergebnissen nützlich.

Quiet-Modus

Mit

`-q`

beziehungsweise

`--quiet`

wird die normale Ausgabe auf `stdout` unterdrückt.

Dies ist beispielsweise bei automatisierter Verarbeitung oder gezielter Protokollierung sinnvoll.

Verbose-Modus

Mit

`-v`

beziehungsweise

`--verbose`

kann eine ausführlichere Ausgabe aktiviert werden.

Bei detaillierten Analysen kann dies zusätzliche Kontextinformationen liefern.

Typisches erstes Vorgehen

Bei einer unbekannten Binärdatei sollte zunächst möglichst wenig verändert beziehungsweise automatisch extrahiert werden.

Ein sinnvoller erster Schritt lautet:

```
binwalk DATEI
```

Damit wird zunächst festgestellt, welche Strukturen erkannt werden.

Danach können interessante Offsets gezielt bewertet werden.

Erst wenn eine Extraktion tatsächlich erforderlich ist, kann:

```
binwalk -e DATEI
```

verwendet werden.

Bei verschachtelten Inhalten kann anschließend eine rekursive Analyse erwogen werden.

Beispiel einer Firmware-Untersuchung

Ein typischer Ablauf kann folgendermaßen aussehen:

1. Originaldatei unverändert sichern.
2. Signaturanalyse durchführen:

```
binwalk firmware.bin
```

3. Erkannte Strukturen und Offsets prüfen.
4. Bei Bedarf Entropie untersuchen:

```
binwalk -E firmware.bin
```

5. Falls eine Extraktion erforderlich ist, ein separates Zielverzeichnis verwenden.
6. Extrahierte Inhalte anschließend separat untersuchen.

Damit bleibt zwischen Original und erzeugten Analysedaten eine klare Trennung erhalten.

Binwalk und Hexedit

Binwalk und Hexedit ergänzen sich unmittelbar.

Binwalk kann beispielsweise melden, dass an einem bestimmten hexadezimalen Offset eine bekannte Datenstruktur beginnt.

Dieser Offset kann anschließend mit Hexedit gezielt untersucht werden.

Die Aufgabenteilung lautet:

Binwalk → Struktur automatisch erkennen

Hexedit → Bytes direkt betrachten und gegebenenfalls bearbeiten

Damit bildet A.15 die automatische strukturelle Analyse und A.16 die manuelle Binäranalyse.

Binwalk und Foremost

Binwalk und Foremost können beide Daten anhand charakteristischer Dateimerkmale erkennen beziehungsweise extrahieren, verfolgen jedoch unterschiedliche Schwerpunkte.

Foremost ist primär ein File-Carving-Werkzeug zur Wiederherstellung beziehungsweise Extraktion bekannter Dateitypen aus Datenströmen und Images.

Binwalk ist primär ein Analysewerkzeug für Binärdateien und Firmware und zeigt insbesondere eingebettete Strukturen und deren Offsets.

Für eine Firmwareanalyse ist Binwalk deshalb wesentlich spezifischer ausgerichtet.

Binwalk und PhotoRec

Auch PhotoRec arbeitet mit Dateisignaturen.

PhotoRec ist jedoch primär auf die Wiederherstellung von Dateien aus Datenträgern und Images ausgerichtet.

Binwalk dient dagegen der Analyse der internen Struktur einer Binärdatei beziehungsweise eines Firmware-Images.

Die Werkzeuge können daher ähnliche technische Grundprinzipien verwenden, erfüllen aber unterschiedliche Rescue-Aufgaben.

Binwalk ist kein Malware-Scanner

Ein erkannter ausführbarer Codebereich oder eine eingebettete Datei bedeutet nicht automatisch, dass Schadsoftware vorhanden ist.

Binwalk klassifiziert Strukturen anhand seiner Analysemechanismen.

Für eine Malware-Untersuchung sind die dafür vorgesehenen Sicherheitswerkzeuge zu verwenden.

Binwalk ist kein Disassembler

Der Opcode-Scan kann Hinweise auf ausführbaren Maschinencode liefern.

Binwalk ersetzt damit jedoch keinen vollständigen Disassembler oder spezialisierten Reverse-Engineering-Debugger.

Seine Stärke liegt in der Identifikation und strukturellen Zerlegung von Binärabbildern.

Binwalk ist kein Datenrettungsprogramm für beschädigte Hardware

Binwalk kann Dateien und Images analysieren.

Es ist jedoch nicht dafür vorgesehen, physisch fehlerhafte Datenträger möglichst schonend auszulesen.

Bei einem beschädigten Datenträger sollte zunächst mit einem dafür vorgesehenen Werkzeug wie GNU ddrescue ein möglichst vollständiges Image erzeugt werden.

Anschließend kann Binwalk gegen diese Kopie beziehungsweise das Image eingesetzt werden.

Originaldateien nicht unnötig verändern

Eine reine Binwalk-Signaturanalyse verändert die untersuchte Datei nicht.

Automatische Extraktion erzeugt dagegen zusätzliche Dateien und Verzeichnisse.

Diese sollten auf einem separaten beschreibbaren Arbeitsmedium abgelegt werden.

Bei forensisch relevanten Untersuchungen sollte das Original unverändert erhalten bleiben.

Nicht vertrauenswürdige Extraktionsinhalte

Extrahierte Dateien stammen aus dem untersuchten Binärabbild.

Sie müssen deshalb grundsätzlich als nicht vertrauenswürdig behandelt werden.

Eine erfolgreiche Extraktion bedeutet nicht, dass die erzeugten Dateien sicher ausgeführt werden können.

Insbesondere ausführbare Programme, Skripte oder unbekannte Binärdateien sollten nicht allein deshalb gestartet werden, weil Binwalk sie erfolgreich extrahiert hat.

Externe Hilfsprogramme

Die automatische Extraktion bestimmter Formate kann von zusätzlichen Programmen abhängig sein.

Fehlt ein benötigtes Hilfsprogramm, kann die Erkennung funktionieren, während die Extraktion scheitert.

Ein solcher Extraktionsfehler bedeutet deshalb nicht zwangsläufig, dass die erkannte Signatur falsch war.

False Positives

Signaturbasierte Analysen können False Positives erzeugen.

Eine zufällige Bytefolge kann einer bekannten Dateisignatur entsprechen.

Binwalk versucht zwar, Ergebnisse anhand seiner Signaturlogik zu bewerten, dennoch sollte ein interessanter Fund durch weitere Analyse bestätigt werden.

Dazu können beispielsweise gehören:

Untersuchung der umgebenden Bytes,

Prüfung der erwarteten Dateistruktur,

erfolgreiche Extraktion,

Vergleich mit bekannten Formaten,

oder eine manuelle Kontrolle mit Hexedit.

Grenzen

Binwalk kann nur Strukturen erkennen, für die entsprechende Erkennungsregeln beziehungsweise Signaturen vorhanden sind oder die durch die verwendeten Analysefunktionen sichtbar werden.

Unbekannte proprietäre Formate können unerkannt bleiben.

Hohe Entropie beweist keine Verschlüsselung.

Ein Signaturtreffer beweist nicht automatisch eine intakte Datei.

Eine erfolgreiche Extraktion beweist nicht die Vertrauenswürdigkeit des Inhalts.

Und ein fehlender Treffer bedeutet nicht, dass eine Binärdatei keine eingebetteten Daten enthält.

Sicherheitshinweise

Originaldateien und Images sollten für Analysezwecke möglichst unverändert erhalten bleiben.

Extrahierte Daten sollten auf einem separaten Arbeitsmedium gespeichert werden.

Automatisch extrahierte Inhalte sind grundsätzlich als nicht vertrauenswürdig zu behandeln.

Rekursive Extraktion kann große Datenmengen erzeugen und externe Hilfsprogramme aufrufen.

Bei unbekannten Dateien sollte deshalb zunächst eine reine Signaturanalyse durchgeführt werden.

Binwalk darf nicht als Ersatz für Malware-Analyse oder Datenträgerrettung verstanden werden.

Bei physisch beschädigten Datenträgern sollte zuerst ein Image mit einem geeigneten Rettungswerkzeug erstellt werden.

Technische Einordnung in JUMBO BLOCK Rescue

Binwalk eröffnet innerhalb von JUMBO BLOCK Rescue den Bereich **Analyse**.

Während die vorherigen Werkzeuge primär:

Daten retten,

Datenträger verwalten,

Hardwarezustände prüfen,

Malware suchen

oder Netzwerkzugriffe kontrollieren,

untersucht Binwalk den **internen strukturellen Aufbau binärer Dateien und Images**.

Seine Kernfunktionen sind:

Signaturanalyse → eingebettete Strukturen identifizieren

Offsetanalyse → Positionen innerhalb einer Binärdatei bestimmen

Extraktion → erkannte Inhalte herauslösen

rekursive Analyse → verschachtelte Strukturen weiter untersuchen

Entropieanalyse → Bereiche mit unterschiedlicher statistischer Datenstruktur sichtbar machen

Zusammen mit A.16 Hexedit entsteht damit eine zweistufige Analyseumgebung:

Binwalk findet automatisch interessante Strukturen und Offsets.

Hexedit ermöglicht anschließend die direkte Untersuchung der betreffenden Bytes.

Binwalk ist damit innerhalb von JUMBO BLOCK Rescue das spezialisierte Werkzeug für die strukturelle Analyse und Zerlegung von Firmware, Binärdateien und vergleichbaren Images.

A.16 Hexedit

Bereich: Analyse
Programm: Hexedit
Baseline 20261001: Hexedit 1.6
Installiertes Paket: `hexedit 1.6-2`
Bedienung: Terminalbasierte interaktive Oberfläche
Aufgabe: Direkte Anzeige, Untersuchung und Bearbeitung von Dateien und Datenträgerinhalten auf Byte-Ebene

Zweck

Hexedit ist ein interaktiver Hexadezimaleditor für die Kommandozeile beziehungsweise das Terminal.

Das Programm stellt den Inhalt einer Datei unmittelbar als Bytefolge dar. Dabei werden die Daten sowohl:

hexadezimal

als auch

als Zeichen

angezeigt.

Hexedit eignet sich dadurch zur direkten Untersuchung binärer Dateien, Images und – mit entsprechenden Zugriffsrechten – Gerätedateien.

Im Unterschied zu Binwalk interpretiert Hexedit den Inhalt nicht primär anhand bekannter Dateisignaturen.

Hexedit stellt die tatsächlich vorhandenen Bytes unmittelbar dar.

Damit beantwortet das Programm insbesondere die Frage:

„Welche Bytes befinden sich an dieser konkreten Position?“

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue dient Hexedit zur manuellen Low-Level-Analyse.

Typische Aufgaben sind:

Untersuchung unbekannter Binärdateien,

Kontrolle von Datei-Headern,

Untersuchung bestimmter Offsets,

Kontrolle eingebetteter Zeichenketten,

Vergleich mit Ergebnissen von Binwalk,

Analyse von Images,

Kontrolle von Dateisystem- oder Partitionsstrukturen durch erfahrene Anwender,

und gezielte Änderung einzelner Bytes, wenn dies ausdrücklich erforderlich ist.

Gerade die letzte Funktion macht Hexedit zu einem leistungsfähigen, aber potenziell gefährlichen Werkzeug.

Grundlegender Aufruf

Die grundlegende Syntax lautet:

```
hexedit [Optionen] [Datei]
```

Beispiel:

```
hexedit firmware.bin
```

Hexedit öffnet die angegebene Datei in seiner terminalbasierten Oberfläche.

Wird eine Datei mit Schreibzugriff geöffnet, können Änderungen vorgenommen und anschließend gespeichert werden.

Bildschirmaufbau

Die Darstellung besteht im Wesentlichen aus drei Informationsbereichen:

Offset

zeigt die Position der dargestellten Daten innerhalb der Datei.

Hexadezimale Darstellung

zeigt die einzelnen Bytewerte als zweistellige Hexadezimalzahlen.

Beispielsweise:

```
4A 55 4D 42 4F
```

ASCII-/Zeichendarstellung

zeigt – soweit darstellbar – die entsprechenden Zeichen.

Damit können sowohl maschinenorientierte Bytewerte als auch lesbare Textbestandteile innerhalb einer Binärdatei erkannt werden.

Hexadezimale Werte

Ein Byte besitzt 256 mögliche Werte.

Hexadezimal werden diese von:

```
00
```

bis:

```
FF
```

dargestellt.

Ein Byte benötigt damit genau zwei Hexadezimalziffern.

Diese Darstellung ist für Binäranalysen wesentlich übersichtlicher als eine lange Folge dezimaler Bytewerte.

Offset

Der Offset bezeichnet die Position eines Bytes relativ zum Beginn der geöffneten Datei beziehungsweise des geöffneten Datenobjekts.

Der erste Bytewert befindet sich am Offset:

`0`

beziehungsweise hexadezimal:

`0x0`.

Offsets sind besonders wichtig, wenn ein anderes Analysewerkzeug eine interessante Struktur an einer bestimmten Position gefunden hat.

Binwalk gibt beispielsweise die Offsets erkannter Strukturen aus.

Diese Position kann anschließend mit Hexedit direkt untersucht werden.

Navigation

Hexedit ermöglicht die Navigation durch die geöffnete Datei mit den üblichen Bewegungstasten.

Dazu gehören insbesondere die Pfeiltasten.

Darüber hinaus stehen Navigationstasten für größere Bewegungen durch die Datei zur Verfügung.

Dadurch kann sowohl byteweise als auch über größere Bereiche navigiert werden.

Zum Dateianfang springen

Mit:

`Ctrl+A`

kann zum Anfang der Datei gesprungen werden.

Dies ist insbesondere für die Untersuchung eines Datei-Headers nützlich.

Viele Dateiformate speichern charakteristische Kennungen und Metadaten unmittelbar am Anfang der Datei.

Zum Dateiende springen

Mit:

`Ctrl+E`

kann zum Ende der Datei gesprungen werden.

Damit lassen sich beispielsweise Trailer, abschließende Datenstrukturen oder angehängte Daten untersuchen.

Zu einem bestimmten Offset springen

Eine besonders wichtige Funktion ist:

`Ctrl+G`

Damit kann eine bestimmte Position innerhalb der Datei direkt angesprungen werden.

Dies ist für die Zusammenarbeit mit Binwalk besonders relevant.

Meldet Binwalk beispielsweise eine interessante Struktur an einem bestimmten Offset, muss nicht manuell durch die gesamte Datei gescrollt werden.

Der Offset kann direkt in Hexedit aufgerufen werden.

Dezimale und hexadezimale Positionen

Bei Low-Level-Analysen muss immer beachtet werden, in welchem Zahlensystem ein Offset angegeben ist.

Ein Wert wie:

`4096`

dezimal

entspricht:

`0x1000`

hexadezimal.

Analysewerkzeuge können Offsets in unterschiedlichen Darstellungen ausgeben.

Vor einer manuellen Änderung muss deshalb eindeutig feststehen, welche Position tatsächlich gemeint ist.

Suche

Hexedit besitzt eine Suchfunktion.

Mit:

`Ctrl+S`

kann nach Daten innerhalb der geöffneten Datei gesucht werden.

Die Suche kann für unterschiedliche Analyseaufgaben eingesetzt werden, beispielsweise zum Auffinden einer bekannten Zeichenfolge oder einer bekannten Bytefolge.

Suche nach Zeichenketten

Eine Datei kann trotz ihres binären Aufbaus lesbare Texte enthalten.

Beispiele können sein:

Versionsangaben,

Herstellerkennungen,

Konfigurationswerte,

Pfade,

Fehlermeldungen

oder andere Textinformationen.

Die Suche ermöglicht es, solche Zeichenfolgen gezielt aufzufinden.

Suche nach Hexadezimalwerten

Hexedit kann ebenfalls zur Suche nach hexadezimal angegebenen Bytefolgen verwendet werden.

Dies ist relevant, wenn nicht nach einem lesbaren Text, sondern nach einer bekannten binären Signatur gesucht wird.

Eine bekannte Magic-Byte-Folge kann dadurch direkt lokalisiert werden.

Suche fortsetzen

Nach einer Suche kann die Suche nach dem nächsten passenden Vorkommen fortgesetzt werden.

Damit lassen sich mehrere identische beziehungsweise ähnliche Strukturen innerhalb einer Datei untersuchen.

Bei großen Images ist dies wesentlich effizienter als ein manuelles Durchscrollen.

Direkte Bearbeitung

Hexedit ist nicht nur ein Viewer.

Das Programm ermöglicht die direkte Änderung der dargestellten Daten.

Dabei kann sowohl im hexadezimalen Bereich als auch im Zeichenbereich gearbeitet werden.

Änderungen wirken auf die entsprechenden Bytes der geöffneten Datei.

Das ist eine wesentliche Eigenschaft des Programms und gleichzeitig sein größtes Risiko.

Hexadezimale Bearbeitung

Wird im Hexbereich gearbeitet, können Bytewerte direkt eingegeben beziehungsweise verändert werden.

Beispielsweise kann:

durch:

42

ersetzt werden.

Damit wird exakt ein Bytewert verändert.

Eine solche Änderung kann bereits ausreichen, um:

eine Prüfsumme ungültig zu machen,

einen Datei-Header zu beschädigen,

eine Struktur unlesbar zu machen,

eine Programmfunktion zu verändern,

oder ein Dateisystem beziehungsweise Image zu beschädigen.

Zeichenbasierte Bearbeitung

Im Zeichenbereich können darstellbare Bytes als Zeichen bearbeitet werden.

Eine Änderung des sichtbaren Zeichens verändert dabei ebenfalls den zugrunde liegenden Bytewert.

Die Zeichenansicht ist deshalb keine separate Textkopie.

Hex- und Zeichenansicht repräsentieren dieselben Daten.

Umschalten zwischen Hex- und Zeichenbereich

Mit:

Tab

kann zwischen der hexadezimalen und der ASCII-/Zeichenbearbeitung gewechselt werden.

Damit lässt sich abhängig von der Aufgabe entweder direkt mit Bytewerten oder mit lesbaren Zeichen arbeiten.

Änderungen sind zunächst editierbar

Während einer Bearbeitung werden Daten innerhalb der Hexedit-Sitzung verändert.

Entscheidend ist anschließend, ob diese Änderungen tatsächlich in die Datei geschrieben werden.

Damit besteht eine wichtige Trennung zwischen:

Änderung im Editor

und

Speicherung auf dem Datenträger.

Vor dem Speichern muss deshalb nochmals geprüft werden, ob tatsächlich das richtige Datenobjekt und der richtige Offset bearbeitet wurden.

Änderungen speichern

Hexedit ermöglicht das Speichern vorgenommener Änderungen.

Eine gespeicherte Änderung verändert die zugrunde liegende Datei beziehungsweise das geöffnete beschreibbare Datenobjekt.

Bei einer normalen Arbeitskopie kann dies beabsichtigt sein.

Bei einem Originaldatenträger oder forensischen Image kann es dagegen unerwünscht oder zerstörerisch sein.

Beenden

Hexedit kann über seine vorgesehene Beenden-Funktion verlassen werden.

Sind Änderungen vorhanden, muss bewusst entschieden werden, ob diese gespeichert oder verworfen werden sollen.

Ein Analysevorgang sollte deshalb nicht mit einem unreflektierten Speichern beendet werden.

Read-only-Modus

Für Rescue- und Analysearbeiten ist die Option:

`-r`

besonders wichtig.

Sie öffnet die Datei im **Read-only-Modus**.

Beispiel:

```
hexedit -r image.dd
```

Damit kann die Datei untersucht werden, ohne sie über Hexedit zu verändern.

Für eine reine Analyse ist dies der bevorzugte Betriebsmodus.

Read-only als Standard für Untersuchungen

Wenn der Zweck ausschließlich lautet:

Daten ansehen,

Offsets prüfen,

Header untersuchen,

Signaturen vergleichen

oder Binwalk-Ergebnisse kontrollieren,

sollte die Datei möglichst mit:

```
hexedit -r DATEI
```

geöffnet werden.

Damit wird das Risiko einer versehentlichen Änderung erheblich reduziert.

Geräte und Datenträger

Hexedit kann unter Unix/Linux nicht nur normale Dateien, sondern grundsätzlich auch zugängliche Gerätedateien öffnen.

Ein Blockgerät kann beispielsweise unter:

```
/dev/...
```

repräsentiert sein.

Damit ist eine unmittelbare Untersuchung von Rohdaten eines Datenträgers technisch möglich.

Dies ist jedoch erheblich riskanter als die Untersuchung einer normalen Arbeitskopie.

Direkter Schreibzugriff auf Blockgeräte

Wird ein Blockgerät beschreibbar geöffnet und anschließend verändert, wirken gespeicherte Änderungen unmittelbar auf die Daten des Datenträgers.

Ein einziges falsch geändertes Byte kann kritische Strukturen beschädigen.

Betroffen sein können beispielsweise:

Partitionsinformationen,

Bootstrukturen,

Dateisystem-Metadaten,

Dateiinhalte

oder andere Verwaltungsdaten.

Direkte Änderungen an einem Originaldatenträger sollten deshalb nur von Anwendern durchgeführt werden, die Aufbau und Folgen der betreffenden Struktur exakt kennen.

Image statt Original

Für Rescue- und Analysearbeiten ist die sicherere Vorgehensweise:

Originaldatenträger sichern → Image beziehungsweise Arbeitskopie erstellen → Kopie analysieren.

Bei einem physisch problematischen Datenträger kann GNU ddrescue verwendet werden, um zunächst ein möglichst vollständiges Image zu erzeugen.

Dieses Image kann anschließend mit Hexedit untersucht werden.

Beispiel:

```
hexedit -r rescue.img
```

Damit wird nicht ständig auf dem beschädigten Originalmedium gearbeitet.

Physisch beschädigte Datenträger

Hexedit ist kein Datenrettungswerkzeug für fehlerhafte Hardware.

Beim Scrollen, Suchen und Navigieren werden Daten vom Datenträger gelesen.

Auf einem physisch ausfallenden Medium können solche zusätzlichen Lesezugriffe unerwünscht sein.

Bei Lesefehlern oder erkennbaren Hardwareproblemen sollte deshalb zuerst eine kontrollierte Sicherung mit einem dafür vorgesehenen Werkzeug wie GNU ddrescue erfolgen.

Hexedit wird anschließend auf das erzeugte Image angewendet.

Binwalk und Hexedit

Binwalk und Hexedit bilden innerhalb von JUMBO BLOCK Rescue ein unmittelbar zusammengehöriges Analysewerkzeugpaar.

Binwalk kann beispielsweise feststellen:

„An Offset X beginnt eine bekannte Datenstruktur.“

Hexedit kann anschließend verwendet werden, um:

die Bytes an Offset X anzusehen,

den Bereich vor und hinter der Signatur zu untersuchen,

Headerwerte manuell zu kontrollieren,

oder die Binwalk-Erkennung unabhängig nachzuvollziehen.

Die Arbeitsteilung lautet:

Binwalk → automatische Strukturerkennung

Hexedit → manuelle Byteanalyse

Beispiel eines Analyseablaufs

Eine unbekannte Firmwaredatei wird zunächst mit Binwalk untersucht:

```
binwalk firmware.bin
```

Binwalk meldet interessante Strukturen und Offsets.

Anschließend wird die Datei sicherheitshalber schreibgeschützt mit Hexedit geöffnet:

```
hexedit -r firmware.bin
```

Mit:

`Ctrl+G`

wird zu einem von Binwalk gemeldeten Offset gesprungen.

Dort können die tatsächlichen Bytes und gegebenenfalls lesbare Zeichen direkt untersucht werden.

So lässt sich ein automatischer Analysefund manuell nachvollziehen.

Hexedit und Dateisignaturen

Viele Dateiformate beginnen mit charakteristischen Bytefolgen.

Diese werden häufig als:

Magic Bytes,

Magic Numbers

oder Dateisignaturen

bezeichnet.

Hexedit ermöglicht es, solche Bytefolgen unmittelbar anzusehen.

Beispielsweise kann damit geprüft werden, ob eine Datei tatsächlich mit einer erwarteten Signatur beginnt oder ob Dateiendung und tatsächlicher Binärinhalt voneinander abweichen.

Beschädigte Datei-Header

Bei beschädigten Dateien kann Hexedit verwendet werden, um den Header direkt zu untersuchen.

Dadurch lässt sich feststellen, ob erwartete Signaturen oder Headerwerte fehlen beziehungsweise verändert sind.

Hexedit repariert einen beschädigten Header jedoch nicht automatisch.

Eine manuelle Reparatur setzt genaue Kenntnis des betreffenden Dateiformats voraus.

Keine automatische Interpretation

Hexedit weiß nicht automatisch, was ein bestimmtes Byte semantisch bedeutet.

Ein Wert wie:

`FF`

kann je nach Dateiformat völlig unterschiedliche Bedeutungen besitzen.

Die Interpretation muss durch den Anwender anhand der jeweiligen Datenstruktur erfolgen.

Das unterscheidet Hexedit von spezialisierten Parsern und Dateisystemwerkzeugen.

Keine automatische Strukturprüfung

Hexedit prüft nicht automatisch, ob eine manuelle Änderung logisch konsistent ist.

Wer beispielsweise einen Längenwert innerhalb eines Headers verändert, muss selbst wissen, ob davon:

Prüfsummen,

Offsets,

Indexstrukturen,

Folgefelder

oder andere Metadaten abhängen.

Der Editor verhindert keine semantisch falschen Änderungen.

Überschreiben statt strukturelles Editieren

Ein Hexeditor arbeitet grundsätzlich auf vorhandenen Bytepositionen.

Die Bearbeitung eines Bytes ist daher nicht mit der komfortablen Bearbeitung eines Textdokuments gleichzusetzen.

Bei Binärformaten können Einfügen, Entfernen oder Verschieben von Daten sämtliche nachfolgenden Offsets verändern.

Hexedit ist primär für die unmittelbare byteorientierte Bearbeitung vorgesehen.

Zeichenkodierung

Die rechte Zeichenansicht darf nicht mit einer vollständigen Textinterpretation verwechselt werden.

Nicht jedes Byte entspricht einem sinnvoll darstellbaren Zeichen.

Mehrbyte-Zeichenkodierungen und komplexe Textformate können in einer einfachen Byte-/Zeichenansicht nicht zwangsläufig so erscheinen, wie sie in einem Texteditor dargestellt würden.

Für die Binäranalyse bleibt deshalb die Hexdarstellung maßgeblich.

Suche ist keine Dateianalyse

Das Auffinden einer Zeichenfolge oder Bytefolge sagt zunächst nur aus, dass diese Datenfolge an der betreffenden Position vorhanden ist.

Daraus lässt sich nicht automatisch ableiten, welche semantische Funktion sie besitzt.

Ein Treffer muss immer im Kontext der umgebenden Datenstruktur bewertet werden.

Keine Malware-Erkennung

Hexedit ist kein Malware-Scanner.

Verdächtige Bytes oder Zeichenketten können manuell untersucht werden, aber Hexedit klassifiziert sie nicht automatisch als Schadsoftware.

Für Malware-Prüfungen stehen innerhalb von JUMBO BLOCK Rescue spezialisierte Werkzeuge wie ClamAV zur Verfügung.

Kein Disassembler

Hexedit zeigt Maschinencode lediglich als Bytes an.

Es übersetzt diese Bytes nicht automatisch in eine vollständige Assemblerdarstellung und analysiert keinen Programmablauf.

Für Reverse Engineering auf Instruktionsebene wären spezialisierte Disassembler beziehungsweise Analysewerkzeuge erforderlich.

Kein File-Carving

Hexedit kann Rohdaten sichtbar machen und durchsuchen.

Es ist jedoch kein automatisches File-Carving-Werkzeug wie PhotoRec oder Foremost.

Werden Dateisignaturen innerhalb eines Images gesucht, können spezialisierte Carving- beziehungsweise Signaturwerkzeuge wesentlich effizienter sein.

Hexedit eignet sich anschließend zur manuellen Kontrolle.

Keine automatische Datenrettung

Hexedit rekonstruiert keine Partitionstabellen.

Es stellt keine gelöschten Dateien automatisch wieder her.

Es korrigiert keine Dateisysteme.

Es liest beschädigte Datenträger nicht fehlertolerant wie GNU ddrescue.

Seine Stärke liegt ausschließlich in der direkten Kontrolle und – falls ausdrücklich gewünscht – Änderung der vorhandenen Bytes.

Typische Rescue-Szenarien

Hexedit kann innerhalb von JUMBO BLOCK Rescue insbesondere verwendet werden für:

manuelle Kontrolle eines Datei-Headers,

Untersuchung eines von Binwalk gemeldeten Offsets,

Suche nach bekannten Zeichenketten,

Suche nach charakteristischen Bytefolgen,

Untersuchung eines Datenträger-Images,

Kontrolle von Rohdaten,

Vergleich ähnlicher Binärdateien anhand bestimmter Positionen,

und in fachkundigen Sonderfällen die gezielte Änderung einzelner Bytes einer Arbeitskopie.

Sicherer Arbeitsablauf

Für eine normale Analyse empfiehlt sich:

1. Originaldatei beziehungsweise Originaldatenträger nicht verändern.
2. Falls erforderlich eine Arbeitskopie oder ein Image erstellen.
3. Hexedit zunächst im Read-only-Modus öffnen:

```
hexedit -r DATEI
```

4. Interessante Positionen über Offsets oder Suche lokalisieren.
5. Daten und Kontext untersuchen.
6. Falls eine Änderung erforderlich ist, ausschließlich eine separate Arbeitskopie beschreibbar öffnen.
7. Änderungen dokumentieren.
8. Das Ergebnis anschließend mit geeigneten Werkzeugen validieren.

Forensische Betrachtung

Bei einer beweisrelevanten Untersuchung sollte ein Original nicht mit einem schreibfähigen Hexeditor geöffnet und verändert werden.

Eine Arbeitskopie sollte klar vom Original getrennt sein.

Zusätzlich können Hashwerte vor und nach einer Analyse beziehungsweise Bearbeitung verwendet werden, um Veränderungen nachvollziehbar zu dokumentieren.

Hexedit selbst ersetzt jedoch keine vollständige forensische Beweismittelkette.

Grenzen

Hexedit interpretiert komplexe Datenstrukturen nicht automatisch.

Der Anwender muss wissen, was die untersuchten Bytes bedeuten.

Das Programm verhindert keine logisch falschen Änderungen.

Eine Byteänderung kann eine gesamte Datei oder Datenstruktur unbrauchbar machen.

Hexedit ersetzt weder einen Dateisystem-Reparaturdienst noch einen Malware-Scanner, Disassembler, Datenretter oder spezialisierten Firmware-Parser.

Sicherheitshinweise

Für reine Untersuchungen sollte Hexedit mit:

`-r`

im Read-only-Modus verwendet werden.

Originaldatenträger sollten nicht unnötig beschreibbar geöffnet werden.

Vor manuellen Änderungen sollte eine Sicherung beziehungsweise Arbeitskopie vorhanden sein.

Direkte Änderungen an `/dev/...`-Geräten können unmittelbar Daten zerstören.

Bei physisch beschädigten Datenträgern sollte zuerst mit einem spezialisierten Rettungswerkzeug ein Image erzeugt werden.

Offsets müssen vor einer Änderung eindeutig überprüft werden.

Besondere Aufmerksamkeit gilt der Unterscheidung zwischen dezimalen und hexadezimalen Positionsangaben.

Eine erfolgreiche Änderung einzelner Bytes bedeutet nicht, dass die gesamte Datenstruktur anschließend konsistent ist.

Technische Einordnung in JUMBO BLOCK Rescue

Hexedit bildet gemeinsam mit Binwalk den Bereich **Analyse** innerhalb von JUMBO BLOCK Rescue.

Die beiden Programme ergänzen sich:

A.15 Binwalk

automatische Analyse binärer Dateien und Images auf bekannte Strukturen, Signaturen, eingebettete Inhalte und interessante Offsets.

A.16 Hexedit

direkte manuelle Untersuchung der tatsächlichen Bytes an diesen Positionen.

Damit ergibt sich eine klare Analysekette:

Binwalk findet.

Hexedit zeigt.

Der Anwender bewertet.

Falls eine Änderung technisch erforderlich ist, kann Hexedit diese ebenfalls durchführen. Genau deshalb ist jedoch der Read-only-Modus für die reine Analyse besonders wichtig.

Hexedit stellt innerhalb von JUMBO BLOCK Rescue somit die niedrigste manuelle Analyseebene der beiden Programme dar:

den unmittelbaren Zugriff auf die Bytes einer Datei oder eines Images.

A.17 Nmap

Bereich: Netzwerk
Programm: Nmap – Network Mapper
Baseline 20261001: Nmap 7.95
Installierte Pakete: `nmap 7.95+dfsg-3`, `nmap-common 7.95+dfsg-3`
Bedienung: Kommandozeile
Aufgabe: Netzwerkerkennung, Portanalyse, Dienst- und Versionserkennung, Betriebssystemerkennung und erweiterte Netzwerkanalyse

Zweck

Nmap steht für:

Network Mapper

Nmap ist ein Werkzeug zur Untersuchung und Inventarisierung von IP-Netzwerken.

Seine ursprüngliche und weiterhin zentrale Aufgabe ist das Port-Scanning. Darüber hinaus kann Nmap unter anderem:

aktive Hosts erkennen,

TCP- und UDP-Ports untersuchen,

Portzustände klassifizieren,

Netzwerkdienste identifizieren,

Softwareversionen von Diensten ermitteln,

Betriebssysteme anhand ihres Netzwerkverhaltens abschätzen,

Netzwerkpfade untersuchen,

und über die Nmap Scripting Engine zusätzliche Netzwerkprüfungen durchführen.

Damit gehört Nmap zu den umfangreichsten Netzwerkdiagnosewerkzeugen von JUMBO BLOCK Rescue.

Einsatzgebiet im Rescue-System

In einer Rescue-Umgebung kann Nmap beispielsweise verwendet werden, um:

erreichbare Geräte in einem lokalen Netz festzustellen,

einen unbekannten Netzwerkbereich zu inventarisieren,

offene Ports eines Systems zu bestimmen,

festzustellen, welcher Netzwerkdienst auf einem Port antwortet,

Versionsinformationen eines Dienstes zu ermitteln,

unerwartet erreichbare Dienste zu entdecken,

Firewall-Auswirkungen zu untersuchen,

oder ein Gerät anhand seines Netzwerkverhaltens näher zu klassifizieren.

Nmap arbeitet dabei aktiv.

Das Programm sendet abhängig vom gewählten Scanverfahren gezielt Netzwerkpakete beziehungsweise Verbindungsversuche an die untersuchten Systeme.

Rechtliche und betriebliche Voraussetzung

Nmap darf nur gegen Systeme und Netzwerke eingesetzt werden, für deren Untersuchung eine entsprechende Berechtigung besteht.

Ein Portscan ist keine rein passive Beobachtung.

Abhängig von den verwendeten Optionen können zahlreiche Pakete, vollständige Verbindungen oder anwendungsspezifische Anfragen erzeugt werden.

Einige NSE-Skripte gehen erheblich weiter und können intrusive Tests, Brute-Force-Prüfungen, Exploit-Tests oder sogar Denial-of-Service-Funktionen enthalten.

In Produktions-, KRITIS-, Industrie- und Fremdnetzen muss deshalb vor einem Scan geklärt sein, welche Untersuchungen zulässig sind.

Grundsyntax

Die grundlegende Syntax lautet:

```
nmap [Optionen] ZIEL
```

Das Ziel kann beispielsweise als:

Hostname,

IPv4-Adresse,

IPv6-Adresse,

Adressbereich

oder Netzwerk

angegeben werden.

Beispiel für einen einzelnen autorisierten Host:

```
nmap 192.0.2.10
```

Die Adresse dient hier ausschließlich als Dokumentationsbeispiel.

Standardscan

Ein einfacher Aufruf:

```
nmap ZIEL
```

führt einen Portscan gegen das Ziel durch.

Nmap scannt standardmäßig die **1.000 am häufigsten verwendeten Ports** des jeweiligen gescannten Protokolls.

Das ist nicht dasselbe wie ein Scan aller 65.535 TCP-Ports.

Host Discovery

Bevor Nmap einen umfangreichen Portscan durchführt, versucht es normalerweise zunächst festzustellen, ob ein Zielhost erreichbar beziehungsweise aktiv ist.

Dieser Vorgang wird als:

Host Discovery

bezeichnet.

Dadurch muss Nmap nicht jeden möglicherweise inaktiven Host vollständig scannen.

Nur Host Discovery

Mit:

`-sn`

wird der anschließende Portscan deaktiviert.

Beispiel:

```
nmap -sn 192.0.2.0/24
```

Nmap führt damit Host Discovery durch und meldet die dabei als erreichbar erkannten Hosts, ohne anschließend einen normalen Portscan auszuführen.

Diese Funktion eignet sich für eine erste Bestandsaufnahme eines autorisierten Netzwerksegments.

„Ping Scan“ bedeutet nicht nur ICMP

Die Option `-sn` wird traditionell auch als Ping Scan bezeichnet.

Das bedeutet jedoch nicht, dass Nmap ausschließlich einen klassischen ICMP-Echo-Request verwendet.

Nmap kann für Host Discovery unterschiedliche Verfahren einsetzen.

Die tatsächlich verwendeten Discovery-Probes hängen unter anderem vom Ziel und von den verfügbaren Rechten ab.

Lokales Ethernet

Bei direkt angeschlossenen Ethernet-Netzen verwendet Nmap für lokale Ziele standardmäßig ARP beziehungsweise bei IPv6 entsprechende Neighbor-Discovery-Mechanismen, weil diese Verfahren für lokale Hosts besonders zuverlässig sind.

Das ist für Rescue-Einsätze wichtig:

Ein Host kann auf ICMP-Ping nicht reagieren und trotzdem von Nmap im lokalen Ethernet erkannt werden.

Host Discovery überspringen

Mit:

`-Pn`

wird die normale Host Discovery übersprungen.

Nmap behandelt die angegebenen Zieladressen dann so, als seien sie aktiv, und führt die angeforderten weiteren Scans aus.

Beispiel:

```
nmap -Pn 192.0.2.10
```

Diese Option kann sinnvoll sein, wenn ein Host Discovery-Probes blockiert, obwohl seine Dienste erreichbar sind.

Konsequenz von `-Pn`

`-Pn` bedeutet nicht:

„Host ist nachweislich online.“

Es bedeutet:

„Nmap soll für die weitere Untersuchung davon ausgehen, dass der Host online ist.“

Bei großen Adressbereichen kann dies einen Scan erheblich verlängern, weil Nmap auch nicht vorhandene beziehungsweise nicht erreichbare Zieladressen untersucht.

Ziele auflisten ohne Scan

Mit:

`-sL`

kann ein List Scan durchgeführt werden.

Dabei listet Nmap die vorgesehenen Ziele auf, ohne sie mit einem normalen Host- oder Portscan zu untersuchen.

Dies ist eine nützliche Kontrollmöglichkeit vor einem größeren Scan.

Damit kann insbesondere geprüft werden, ob der angegebene Adressbereich tatsächlich dem beabsichtigten Zielbereich entspricht.

Ports

TCP und UDP verwenden numerische Portnummern.

Der reguläre Portbereich reicht von:

`1`

bis:

`65535`.

Portnummern helfen dabei, verschiedene Netzwerkdienste eines Hosts voneinander zu unterscheiden.

Ein offener Port zeigt jedoch zunächst nur, dass über das betreffende Protokoll ein Dienst beziehungsweise eine Anwendung erreichbar erscheint.

Standardmäßig 1.000 Ports

Ohne eine spezielle Portangabe scannt Nmap normalerweise die 1.000 am häufigsten verwendeten Ports.

Die Auswahl basiert auf Häufigkeitsinformationen in der Nmap-Dienstedatenbank.

Dadurch ist ein Standardsuchlauf wesentlich schneller als ein vollständiger Scan aller Ports.

Bestimmte Ports auswählen

Mit:

`-p`

können bestimmte Ports vorgegeben werden.

Beispiel:

```
nmap -p 22,80,443 192.0.2.10
```

Damit werden nur die angegebenen Ports untersucht.

Auch Bereiche können angegeben werden:

```
nmap -p 1-1024 192.0.2.10
```

Alle Ports

Mit:

`-p-`

werden die Ports 1 bis 65535 gescannt.

Beispiel:

```
nmap -p- 192.0.2.10
```

Ein vollständiger Portscan erzeugt erheblich mehr Netzwerkverkehr und benötigt entsprechend mehr Zeit als ein Standardsuchlauf.

Fast Scan

Mit:

`-F`

wird die Zahl der untersuchten Ports reduziert.

Bei vorhandenen Häufigkeitsinformationen in `nmap-services` werden damit die **100 häufigsten Ports** statt der standardmäßigen 1.000 untersucht.

Dies eignet sich für eine schnelle erste Übersicht.

Top Ports

Mit:

`--top-ports N`

kann die Zahl der anhand ihrer Häufigkeit ausgewählten Ports explizit angegeben werden.

Beispiel:

```
nmap --top-ports 20 192.0.2.10
```

Damit werden die 20 nach der Nmap-Datenbasis am höchsten gewichteten Ports untersucht.

TCP und UDP unterscheiden

TCP- und UDP-Ports sind getrennte Namensräume.

`53/tcp`

und

`53/udp`

sind deshalb zwei unterschiedliche Untersuchungsziele.

Eine vollständige Netzwerkbewertung darf UDP-Dienste nicht automatisch ausblenden.

Die sechs Portzustände

Nmap unterscheidet sechs Portzustände:

`open`

`closed`

`filtered`

`unfiltered`

`open|filtered`

`closed|filtered`

Diese Zustände beschreiben nicht zwingend eine absolute Eigenschaft des Ports.

Sie beschreiben, **wie Nmap den Port aus Sicht des aktuellen Scanpunkts und anhand der erhaltenen Antworten beurteilt.**

open

`open` bedeutet, dass eine Anwendung aktiv TCP-Verbindungen, UDP-Datagramme oder entsprechende Protokollkommunikation auf diesem Port annimmt.

Ein offener Port ist nicht automatisch eine Sicherheitslücke.

Er zeigt zunächst, dass dort ein Netzwerkdienst erreichbar ist.

Anschließend muss bewertet werden:

Welcher Dienst ist das?

Wird er benötigt?

Ist er korrekt abgesichert?

Soll er aus diesem Netzwerk erreichbar sein?

closed

`closed` bedeutet, dass der Port erreichbar ist und auf Nmaps Testpakete reagiert, dort jedoch keine Anwendung auf Verbindungen lauscht.

Ein geschlossener Port liefert damit ebenfalls Informationen:

Der Zielhost beziehungsweise Netzwerkstack ist grundsätzlich erreichbar.

filtered

`filtered` bedeutet, dass Nmap nicht feststellen kann, ob der Port offen ist, weil ein Paketfilter oder anderes Netzwerkhindernis die Untersuchung verhindert.

Ursache kann beispielsweise sein:

eine hostbasierte Firewall,

eine externe Firewall,

eine Router-Regel,

oder ein anderes Filterelement.

`filtered` bedeutet deshalb nicht automatisch:

„Port geschlossen.“

unfiltered

`unfiltered` bedeutet, dass der Port erreichbar ist, Nmap mit dem verwendeten Scanverfahren jedoch nicht bestimmen kann, ob er offen oder geschlossen ist.

Dieser Zustand tritt insbesondere beim ACK-Scan auf.

`open|filtered`

`open|filtered` bedeutet, dass Nmap nicht unterscheiden kann, ob ein Port offen oder gefiltert ist.

Dies tritt bei Scanverfahren auf, bei denen ein offener Port möglicherweise keine Antwort erzeugt.

Ein typisches Beispiel ist der UDP-Scan.

`closed|filtered`

`closed|filtered` bedeutet, dass Nmap nicht bestimmen kann, ob ein Port geschlossen oder gefiltert ist.

Dieser Zustand wird beim IP-ID-Idle-Scan verwendet.

Ergebnisse sind perspektivabhängig

Die Nmap-Portzustände gelten aus Sicht des Systems, von dem aus der Scan durchgeführt wurde.

Ein Port kann beispielsweise:

innerhalb eines lokalen Netzes `open`

und

aus einem anderen Netzwerk `filtered`

erscheinen.

Firewall-Regeln, Routing, NAT und andere Netzwerkelemente beeinflussen das Ergebnis.

TCP SYN Scan

Die Option:

`-sS`

aktiviert den TCP-SYN-Scan.

Beispiel:

```
sudo nmap -sS 192.0.2.10
```

Dieser Scan wird häufig als **Half-open Scan** bezeichnet.

Nmap sendet ein SYN-Paket, wie es normalerweise zum Beginn eines TCP-Verbindungsaufbaus verwendet wird.

Antwortet der Zielport mit SYN/ACK, deutet dies auf einen offenen Port hin.

Antwortet er mit RST, gilt er als geschlossen.

Bei fehlender beziehungsweise bestimmter ICMP-Antwort kann der Port als gefiltert klassifiziert werden.

SYN Scan und Rechte

Der SYN-Scan benötigt unter Unix/Linux die Möglichkeit, Raw Packets zu senden.

Dafür sind entsprechende Privilegien erforderlich.

Stehen diese nicht zur Verfügung, verwendet Nmap für einen normalen TCP-Scan stattdessen typischerweise den TCP-Connect-Scan.

TCP Connect Scan

Die Option:

`-sT`

aktiviert den TCP-Connect-Scan.

Beispiel:

```
nmap -sT 192.0.2.10
```

Dabei verwendet Nmap den `connect()`-Systemaufruf des Betriebssystems.

Bei einem offenen Port wird dadurch eine vollständige TCP-Verbindung aufgebaut.

Unterschied zwischen `-sS` und `-sT`

Beim SYN-Scan wird der vollständige TCP-Verbindungsaufbau normalerweise nicht abgeschlossen.

Beim Connect-Scan wird die Verbindung über das Betriebssystem tatsächlich aufgebaut.

Der Connect-Scan benötigt keine Raw-Packet-Rechte, ist aber typischerweise weniger effizient und kann auf dem Zielsystem deutlicher als normale Verbindungsversuche protokolliert werden.

UDP Scan

Die Option:

`-sU`

aktiviert einen UDP-Portscan.

Beispiel:

```
sudo nmap -sU 192.0.2.10
```

UDP-Scans sind technisch schwieriger als TCP-Scans, weil UDP keinen TCP-ähnlichen Verbindungsaufbau besitzt.

Viele offene UDP-Dienste antworten auf ein ungeeignetes oder leeres Paket überhaupt nicht.

UDP-Ergebnisse

Antwortet ein UDP-Dienst passend, kann Nmap einen Port als offen klassifizieren.

Ein ICMP-Port-Unreachable zeigt typischerweise einen geschlossenen UDP-Port an.

Erhält Nmap keine eindeutige Antwort, kann das Ergebnis:

`open|filtered`

lauten.

Das bedeutet:

Der Port könnte offen sein und nicht antworten – oder ein Filter könnte die Kommunikation unterdrücken.

UDP-Scans können langsam sein

UDP-Scans können wesentlich länger dauern als TCP-Scans.

Ein Grund dafür ist, dass fehlende Antworten und ICMP-Rate-Limits Wartezeiten und Wiederholungen verursachen.

Für Rescue-Diagnosen ist es daher häufig sinnvoll, zunächst gezielt relevante UDP-Ports zu untersuchen.

TCP und UDP kombinieren

UDP-Scanning kann mit einem TCP-Scan kombiniert werden.

Beispiel:

```
sudo nmap -sS -sU -p T:22,80,443,U:53,161 192.0.2.10
```

Dabei kennzeichnet:

`T:`

TCP-Ports

und

`U:`

UDP-Ports.

ACK Scan

Mit:

`-sA`

führt Nmap einen TCP-ACK-Scan durch.

Dieser Scan ist nicht dafür vorgesehen festzustellen, ob ein Port offen ist.

Er dient insbesondere dazu, Informationen über Filter- beziehungsweise Firewall-Verhalten zu gewinnen.

Er kann Ports beispielsweise als:

`filtered`

oder

`unfiltered`

klassifizieren.

Weitere Scanverfahren

Nmap 7.95 unterstützt weitere spezialisierte Scanverfahren.

Dazu gehören unter anderem:

TCP NULL Scan,

FIN Scan,

Xmas Scan,

Window Scan,

Maimon Scan,

Idle Scan,

SCTP-Scans

und IP-Protokoll-Scanning.

Diese Verfahren sind für spezielle Analyseaufgaben vorgesehen.

Sie sollten nicht wahllos eingesetzt werden.

Ihre Aussagekraft hängt unter anderem vom Zielbetriebssystem, dessen Netzwerkstack und vorhandenen Filtern ab.

Service Detection

Ein offener Port sagt zunächst nicht zuverlässig, welcher konkrete Dienst dort läuft.

Die Zuordnung:

Port 22 = SSH

Port 80 = HTTP

Port 443 = HTTPS

ist lediglich eine typische Zuordnung.

Ein Administrator kann einen Dienst grundsätzlich auf einem anderen Port betreiben.

Versions- und Diensterkennung

Mit:

`-sV`

aktiviert Nmap die Service- und Versionserkennung.

Beispiel:

```
nmap -sV 192.0.2.10
```

Nmap sendet dabei zusätzliche Probes an erkannte Ports und analysiert deren Antworten.

Ziel ist es, möglichst genau festzustellen, welcher Dienst tatsächlich erreichbar ist.

Mögliche `-sV`-Informationen

Nmap kann dabei – sofern die Antworten dies ermöglichen – Informationen bestimmen wie:

Dienstprotokoll,

Anwendungsname,

Softwareversion,

Hostname,

Gerätetyp,

Betriebssystemfamilie,

und CPE-Kennung.

Nicht jeder Dienst liefert alle diese Informationen.

`nmap-service-probes`

Die Versionserkennung verwendet die Nmap-Datenbank:

`nmap-service-probes`.

Darin befinden sich Probes für verschiedene Dienste sowie Regeln zur Erkennung und Auswertung ihrer Antworten.

Nmap verlässt sich bei `-sV` daher nicht lediglich auf die Portnummer.

Intensität der Versionserkennung

Mit:

```
--version-intensity
```

kann gesteuert werden, wie intensiv Nmap verschiedene Service-Probes verwendet.

Der Wertebereich reicht von:

`0`

bis:

`9`.

Der dokumentierte Standardwert ist:

7.

Eine höhere Intensität erhöht die Wahrscheinlichkeit einer genauen Identifikation, kann aber länger dauern und mehr Netzwerkkommunikation erzeugen.

Light Version Scan

`--version-light`

entspricht:

`--version-intensity 2`

und führt eine schnellere, weniger intensive Versionserkennung durch.

Alle Version-Probes

`--version-all`

entspricht:

`--version-intensity 9`.

Damit versucht Nmap alle geeigneten Versionserkennungs-Probes.

Dies ist entsprechend umfangreicher.

Betriebssystemerkennung

Mit:

`-O`

aktiviert Nmap die Betriebssystemerkennung.

Beispiel:

```
sudo nmap -O 192.0.2.10
```

Nmap verwendet dabei TCP/IP-Stack-Fingerprinting.

Es sendet verschiedene TCP- und UDP-Probes und untersucht detailliert die Antworten des Zielsystems.

OS-Fingerprint

Nmap vergleicht die beobachteten Netzwerkmerkmale mit seiner Fingerprint-Datenbank:

`nmap-os-db`.

Daraus kann Nmap unter geeigneten Bedingungen Informationen ableiten wie:

Hersteller,

Betriebssystemfamilie,

Betriebssystemgeneration,

Gerätetyp

und teilweise CPE-Informationen.

OS-Erkennung ist eine Schätzung

Die Betriebssystemerkennung ist keine direkte Abfrage des installierten Betriebssystems.

Sie basiert auf dessen beobachtetem Netzwerkverhalten.

Das Ergebnis kann deshalb:

eindeutig,

mehrdeutig

oder nicht bestimmbar

sein.

Besonders gute Bedingungen bestehen laut Nmap, wenn mindestens ein offener und ein geschlossener TCP-Port gefunden wurden.

Aggressivere OS-Schätzung

Mit:

`--osscan-guess`

kann Nmap bei nicht eindeutig passenden Fingerprints aggressiver nahe Übereinstimmungen ausgeben.

Solche Ergebnisse werden mit entsprechenden Wahrscheinlichkeits- beziehungsweise Vertrauensangaben gekennzeichnet.

Eine solche Schätzung darf nicht als sicher festgestelltes Betriebssystem dokumentiert werden.

Aggressive Detection

Die Option:

`-A`

aktiviert mehrere fortgeschrittene Funktionen gemeinsam.

Dazu gehören insbesondere:

Betriebssystemerkennung,

Versionserkennung,

Script-Scanning

und Traceroute.

`-A` bedeutet daher nicht lediglich:

„gründlicher Portscan“.

Es aktiviert zusätzliche aktive Analyseverfahren.

-A bewusst einsetzen

Ein:

```
nmap -A ZIEL
```

erzeugt erheblich mehr Interaktion mit dem Ziel als ein einfacher Portscan.

In sensiblen Produktions- oder KRITIS-Netzen sollte diese Option deshalb nicht reflexartig verwendet werden.

Zuerst sollte geklärt werden, welche Informationen tatsächlich benötigt werden.

Nmap Scripting Engine

Eine besonders leistungsfähige Komponente ist die:

Nmap Scripting Engine – NSE.

NSE verwendet Lua-Skripte, um zahlreiche Netzwerkaufgaben zu automatisieren.

Dazu gehören unter anderem:

erweiterte Netzwerkerkennung,

zusätzliche Serviceerkennung,

Informationsabfragen,

Authentifizierungsprüfungen,

Malware-Prüfungen,

Schwachstellenprüfungen

und weitere spezialisierte Tests.

Standardskripte

Mit:

```
-sC
```

wird der Standard-Satz der NSE-Skripte ausgeführt.

Dies entspricht:

```
--script=default
```

Beispiel:

```
nmap -sC 192.0.2.10
```

NSE ist nicht grundsätzlich passiv oder harmlos

Die offizielle Nmap-Dokumentation weist ausdrücklich darauf hin, dass selbst einige Skripte der Kategorie `default` als intrusive betrachtet werden können.

NSE-Skripte laufen außerdem nicht in einer Sandbox.

Drittanbieter-Skripte sollten deshalb nur verwendet werden, wenn ihre Herkunft und Funktion vertrauenswürdig beziehungsweise geprüft sind.

NSE-Kategorien

Nmap ordnet NSE-Skripte Kategorien zu.

Dazu gehören in Nmap 7.x unter anderem:

`auth`

`broadcast`

`brute`

`default`

`discovery`

`dos`

`exploit`

`external`

`fuzzer`

`intrusive`

`malware`

`safe`

`version`

`vuln`.

Diese Kategorien beschreiben unterschiedliche Einsatzzwecke und teilweise auch das Risikoprofil der Skripte.

`safe` bedeutet nicht risikofrei

Die Kategorie:

`safe`

enthält Skripte, die nicht dafür ausgelegt sind, Dienste zum Absturz zu bringen, Schwachstellen auszunutzen oder große Ressourcenmengen zu verbrauchen.

Das bedeutet dennoch nicht, dass ein solcher Scan unter allen Umständen ohne Nebenwirkungen bleibt.

Auch diese Skripte erzeugen aktive Netzwerkkommunikation.

Intrusive Scripts

Die Kategorie:

`intrusive`

enthält Skripte, bei denen ein höheres Risiko besteht, dass sie:

Ressourcen des Zielsystems beanspruchen,

einen Dienst beeinträchtigen,

oder von Administratoren als Angriff wahrgenommen werden.

Solche Skripte sollten in produktiven Systemen nur nach ausdrücklicher Freigabe eingesetzt werden.

`vuln`

Die Kategorie:

`vuln`

enthält Skripte zur Prüfung auf bestimmte bekannte Schwachstellen.

Das Ergebnis eines solchen Skripts muss technisch bewertet werden.

Ein Skript ersetzt keine vollständige Schwachstellenanalyse.

`exploit`, `brute` und `dos`

Besondere Vorsicht gilt für Kategorien wie:

`exploit`

`brute`

und

`dos`.

Sie können gezielt exploitierende, authentifizierungsbezogene oder dienstbelastende Funktionen enthalten.

Solche Kategorien gehören nicht zu einer normalen Rescue-Netzwerkdiagnose.

Ihre bloße technische Verfügbarkeit ist keine Freigabe zu ihrer Verwendung.

Einzelnes NSE-Skript

Ein bestimmtes Skript kann mit:

`--script`

ausgewählt werden.

Grundprinzip:

```
nmap --script SCRIPTNAME ZIEL
```

Vor der Ausführung sollte dessen Funktion geprüft werden.

Hilfe zu einem NSE-Skript

Mit:

```
--script-help
```

kann die Dokumentation eines Skripts angezeigt werden.

Beispielprinzip:

```
nmap --script-help SCRIPTNAME
```

Dies ist vor dem Einsatz eines unbekannten NSE-Skripts ausdrücklich sinnvoll.

Skriptargumente

Einige NSE-Skripte akzeptieren zusätzliche Parameter.

Diese werden mit:

```
--script-args
```

übergeben.

Die benötigten Parameter sind skriptabhängig und sollten ausschließlich entsprechend der jeweiligen Skriptdokumentation verwendet werden.

Ausgabe

Nmap kann Ergebnisse direkt im Terminal anzeigen oder in Dateien speichern.

Für dokumentierte Rescue-Einsätze ist die Speicherung der Scanergebnisse besonders nützlich.

Normale Ausgabe

Mit:

```
-oN DATEI
```

wird die normale Nmap-Ausgabe gespeichert.

Beispiel:

```
nmap -oN scan.txt 192.0.2.10
```

Dieses Format eignet sich gut für die menschliche Auswertung.

XML-Ausgabe

Mit:

`-oX DATEI`

wird eine XML-Ausgabe erzeugt.

Beispiel:

```
nmap -oX scan.xml 192.0.2.10
```

XML ist insbesondere für die maschinelle Weiterverarbeitung geeignet.

Grepable Output

Mit:

`-oG DATEI`

kann das ältere grepbare Ausgabeformat erzeugt werden.

Die Nmap-Dokumentation bezeichnet dieses Format inzwischen als **deprecated**.

Für neue automatisierte Verarbeitung ist XML die geeignetere strukturierte Ausgabe.

Mehrere Formate gleichzeitig

Mit:

`-oA BASISNAME`

speichert Nmap die Ergebnisse gleichzeitig als:

normale Ausgabe,

XML

und grepbares Format.

Beispiel:

```
nmap -oA netzscan 192.0.2.10
```

Es entstehen entsprechend Dateien mit den Endungen:

`.nmap`

`.xml`

`.gnmap`.

Verbose-Ausgabe

Mit:

`-v`

wird die Ausgabe ausführlicher.

Mehrere v erhöhen die Ausführlichkeit:

`-vv`

Damit zeigt Nmap während des Scans zusätzliche Informationen und beispielsweise gefundene offene Ports früher an.

Nur offene beziehungsweise potenziell offene Ports

Mit:

`--open`

kann die Ausgabe auf Ports beschränkt werden, die mindestens potenziell offen sind.

Dies kann umfangreiche Ergebnisse übersichtlicher machen.

Es verändert jedoch die Darstellung und sollte nicht dazu führen, dass die Existenz anderer Portzustände vergessen wird.

Netzwerkinterfaces und Routen

Mit:

`--iflist`

zeigt Nmap die von ihm erkannten Netzwerkinterfaces und Routinginformationen an und beendet sich anschließend.

Dies ist nützlich, wenn unklar ist, welches Interface beziehungsweise welche Route Nmap für einen Scan verwenden würde.

Timing

Nmap bietet verschiedene Timing-Mechanismen zur Steuerung der Scan-Geschwindigkeit.

Eine bekannte Kurzform sind die Timing Templates:

`-T0`

bis:

`-T5`.

Sie reichen von sehr langsamen bis zu sehr aggressiven Timing-Einstellungen.

Timing Templates nicht mit Scanqualität verwechseln

Ein höheres Timing Template bedeutet nicht automatisch:

„besserer Scan“.

Aggressivere Timings können in schnellen und zuverlässigen Netzen Zeit sparen.

In langsamen, verlustbehafteten oder stark gefilterten Netzen können sie dagegen zu unzuverlässigeren Ergebnissen oder unnötiger Last führen.

Umfang beeinflusst die Laufzeit

Die Scanzeit hängt unter anderem ab von:

Anzahl der Ziele,

Anzahl der Ports,

TCP oder UDP,

Firewalls,

Paketverlust,

Rate Limits,

Versionserkennung,

OS Detection,

NSE-Skripten,

und gewählten Timing-Parametern.

Ein vollständiger UDP-Scan kann beispielsweise erheblich länger dauern als ein kleiner TCP-Scan.

Firewall-Einfluss

Eine Firewall kann Nmap-Ergebnisse wesentlich beeinflussen.

Ein gefilterter Port liefert häufig keine direkte Antwort.

Nmap muss dann gegebenenfalls auf Timeouts warten und Probes wiederholen.

Dadurch können stark gefilterte Netze einen Scan erheblich verlangsamen.

GUFW und Nmap

Die lokale GUFW/UFW-Konfiguration von JUMBO BLOCK Rescue kann ebenfalls Netzwerkkommunikation beeinflussen.

Wenn unerwartete Scanergebnisse auftreten, sollte daher auch geprüft werden:

welches Interface verwendet wird,

welche Route aktiv ist,

und welche lokale Firewall-Konfiguration gilt.

Die Firewall sollte nicht pauschal deaktiviert werden, nur um einen Scan durchzuführen.

Nmap und ARP Scan

A.20 ARP Scan wird speziell auf die Erkennung von Geräten im lokalen Layer-2-Netz ausgerichtet sein.

Nmap besitzt ebenfalls leistungsfähige lokale Host-Discovery-Mechanismen.

Die Werkzeuge überschneiden sich daher teilweise, haben aber unterschiedliche Schwerpunkte.

Nmap bietet zusätzlich umfangreiche Port-, Dienst-, Betriebssystem- und Scriptanalyse.

Nmap und Wireshark

Nmap und Wireshark erfüllen grundsätzlich unterschiedliche Aufgaben.

Nmap erzeugt aktiv Netzwerkprobes und wertet Antworten aus.

Wireshark zeichnet sichtbaren Netzwerkverkehr auf und analysiert Pakete.

Beide Werkzeuge können sich hervorragend ergänzen.

Ein Nmap-Scan kann beispielsweise parallel mit Wireshark beobachtet werden, um die tatsächlich gesendeten und empfangenen Pakete zu untersuchen.

Nmap und TCPDump

TCPDump kann ebenfalls den von Nmap erzeugten Netzwerkverkehr aufzeichnen.

Damit lässt sich insbesondere in einer Terminalumgebung nachvollziehen, welche Pakete während eines Scans übertragen werden.

Nmap und MTR

MTR untersucht primär den Netzwerkpfad und die Erreichbarkeit über mehrere Hops.

Nmap untersucht dagegen primär Hosts, Ports, Dienste und deren Netzwerkmerkmale.

Damit ergänzen sich beide Werkzeuge bei der Diagnose komplexer Netzwerkprobleme.

Nmap und Dig

Dig dient der gezielten DNS-Diagnose.

Nmap kann zwar DNS-bezogene Dienste erkennen und über NSE zusätzliche DNS-Prüfungen durchführen, ersetzt aber kein spezialisiertes DNS-Abfragewerkzeug.

Typischer Rescue-Ablauf

Eine kontrollierte Netzwerkanalyse sollte schrittweise erfolgen.

1. Netzwerkbereich eindeutig bestimmen

Vor jedem Scan muss feststehen, welches Netzwerk untersucht werden darf.

2. Zielbereich kontrollieren

Bei größeren Bereichen kann zunächst ein List Scan sinnvoll sein:

```
nmap -sL NETZ
```

3. Hosts erkennen

Anschließend:

```
nmap -sn NETZ
```

4. Interessanten Host gezielt untersuchen

Beispielsweise:

```
nmap HOST
```

5. Benötigte Ports gezielt untersuchen

```
nmap -p PORTLISTE HOST
```

6. Falls erforderlich Service Detection

```
nmap -sV HOST
```

7. Nur bei begründetem Bedarf weitere Analyse

beispielsweise OS Detection oder ausgewählte NSE-Skripte.

Damit wird nicht automatisch die maximal mögliche Scanintensität gegen jedes Ziel eingesetzt.

Beispiel: Unbekanntes Gerät im lokalen Netz

Wird ein unbekannter Host gefunden, kann zunächst ein normaler Portscan zeigen, welche häufigen TCP-Ports erreichbar sind.

Danach kann:

```
nmap -sV HOST
```

zusätzliche Informationen über die dort antwortenden Dienste liefern.

Bei entsprechender Berechtigung und technischem Bedarf kann anschließend:

```
sudo nmap -O HOST
```

eine Betriebssystemabschätzung versuchen.

Die Ergebnisse werden danach gemeinsam bewertet.

Beispiel: Dienst nicht erreichbar

Ist ein bestimmter Dienst nicht erreichbar, kann gezielt dessen Port untersucht werden:

```
nmap -p PORT HOST
```

Ein Ergebnis:

```
closed
```

bedeutet etwas anderes als:

`filtered`.

`closed` zeigt, dass der Port erreichbar ist, dort aber kein Dienst lauscht.

`filtered` bedeutet, dass Nmap aufgrund einer Filterung keine eindeutige Aussage über offen oder geschlossen treffen kann.

Diese Unterscheidung ist für die Fehlersuche wesentlich.

Beispiel: UDP-Dienst

Für einen bekannten UDP-Dienst sollte gezielt UDP untersucht werden:

```
sudo nmap -sU -p PORT HOST
```

Ein normaler TCP-Scan würde einen ausschließlich über UDP angebotenen Dienst nicht korrekt erfassen.

Scanergebnisse dokumentieren

Bei technischen Einsätzen sollten relevante Scans gespeichert werden.

Beispiel:

```
nmap -sV -oA analyse HOST
```

Damit stehen sowohl eine menschenlesbare als auch strukturierte Ausgaben für spätere Auswertung zur Verfügung.

Zeitpunkt, Scanquelle, Zielbereich und verwendete Optionen sollten ebenfalls dokumentiert werden.

Grenzen

Nmap sieht ein Ziel aus der Netzwerkperspektive des Scanhosts.

Firewall, NAT, Routing und Paketfilter können Ergebnisse beeinflussen.

Ein `open`-Port beweist nicht, dass der Dienst sicher oder unsicher ist.

Ein `filtered`-Port beweist nicht, dass dort kein Dienst läuft.

Service Detection kann sich irren.

OS Detection ist eine Fingerprint-basierte Schätzung.

NSE-Ergebnisse müssen im Kontext des jeweiligen Skripts bewertet werden.

Ein unauffälliger Nmap-Scan beweist nicht die vollständige Sicherheit eines Systems.

Sicherheitshinweise

Nur autorisierte Zielsysteme und Netzwerkbereiche scannen.

Vor größeren Scans Zielbereich und Netzmaske sorgfältig kontrollieren.

Produktions- und KRITIS-Systeme nicht unnötig mit aggressiven Scanoptionen belasten.

UDP-Scans können lange dauern und erheblichen Verkehr erzeugen.

`-A` aktiviert mehrere zusätzliche Analyseverfahren und sollte bewusst eingesetzt werden.

NSE-Skripte vor der Verwendung anhand ihrer Dokumentation und Kategorie prüfen.

Skripte aus den Kategorien `intrusive`, `brute`, `exploit` oder `dos` nicht als normale Diagnosewerkzeuge behandeln.

Drittanbieter-NSE-Skripte nur aus vertrauenswürdiger Quelle und nach Prüfung einsetzen.

Scanergebnisse enthalten potenziell sensible Informationen über die untersuchte Infrastruktur und sollten entsprechend geschützt werden.

Technische Einordnung in JUMBO BLOCK Rescue

Nmap eröffnet innerhalb von JUMBO BLOCK Rescue den umfangreichen Bereich **Netzwerk**.

Seine Kernaufgaben lassen sich in fünf Ebenen gliedern:

1. Host Discovery
Welche Systeme sind erreichbar?

2. Port Scanning
Welche TCP- oder UDP-Ports erscheinen offen, geschlossen oder gefiltert?

3. Service Detection
Welche Netzwerkdienste antworten auf diesen Ports?

4. OS Detection
Welche Betriebssystem- beziehungsweise Geräteklasse passt zum beobachteten Netzwerkverhalten?

5. NSE
Welche zusätzlichen autorisierten Netzwerkprüfungen sollen gezielt durchgeführt werden?

Damit liefert Nmap eine aktive Sicht auf die Netzwerkumgebung.

Die weiteren Netzwerkwerkzeuge von JUMBO BLOCK Rescue ergänzen diese Perspektive:

A.18 Wireshark – grafische Paketaufzeichnung und Protokollanalyse

A.19 TCPDump – terminalbasierte Paketaufzeichnung

A.20 ARP Scan – lokale Layer-2-Geräteerkennung

A.21 MTR – Pfad-, Latenz- und Erreichbarkeitsdiagnose

A.22 Dig – DNS-Diagnose

A.23 Whois – Abfrage von Registrierungsinformationen

A.24 Chromium – webbasierter Zugriff auf Netzwerk- und Administrationsoberflächen

Nmap übernimmt dabei die Rolle des aktiven Netzwerk-Mappers:

Hosts finden → Ports untersuchen → Dienste identifizieren → Systeme klassifizieren → gezielte weiterführende Analyse ermöglichen.

A.18 Wireshark

Bereich: Netzwerk
Programm: Wireshark
Baseline 20261001: Wireshark 4.4.18
Installierte Pakete: `wireshark 4.4.18-0+deb13u1`, `wireshark-common 4.4.18-0+deb13u1`
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Aufzeichnung, Dekodierung und detaillierte Analyse von Netzwerkpaketen

Zweck

Wireshark ist ein Netzwerkprotokollanalysator.

Das Programm kann Netzwerkverkehr aufzeichnen, gespeicherte Paketmitschnitte öffnen und die enthaltenen Netzwerkprotokolle detailliert analysieren.

Während Nmap ein Netzwerk aktiv untersucht, indem gezielt Pakete und Anfragen erzeugt werden, arbeitet Wireshark primär als Beobachtungs- und Analysewerkzeug.

Seine Kernaufgaben sind:

Pakete erfassen

Protokolle dekodieren

Kommunikationsabläufe untersuchen

Pakete filtern

Verbindungen und Datenströme analysieren

Netzwerkstatistiken erzeugen

Capture-Dateien speichern und erneut untersuchen

Damit ist Wireshark innerhalb von JUMBO BLOCK Rescue das zentrale grafische Werkzeug für die Paket- und Protokollanalyse.

Einsatzgebiet im Rescue-System

Wireshark kann insbesondere verwendet werden, um:

Netzwerkkommunikation eines Systems zu beobachten,

Verbindungsprobleme zu analysieren,

DNS-Abfragen zu untersuchen,

TCP-Verbindungsaufbau und -abbau nachzuvollziehen,

DHCP-Kommunikation zu analysieren,

ICMP-Verkehr zu untersuchen,

ARP-Kommunikation sichtbar zu machen,

HTTP- und andere Anwendungsprotokolle zu analysieren,

unerwartete Netzwerkkommunikation zu erkennen,

Nmap-Scans auf Paketebene nachzuvollziehen,

Firewall-Probleme einzugrenzen,

und vorhandene PCAP- beziehungsweise PCAPNG-Dateien offline auszuwerten.

Wireshark ist kein Portscanner

Wireshark und Nmap erfüllen unterschiedliche Aufgaben.

Nmap

erzeugt aktiv Netzwerkprobes und wertet die Antworten aus.

Wireshark

zeichnet Netzwerkpakete auf und dekodiert deren Inhalt.

Wireshark führt daher nicht automatisch einen Portscan durch.

Es kann jedoch den Netzwerkverkehr eines Nmap-Scans sichtbar machen.

Live Capture und Offline-Analyse

Wireshark besitzt zwei grundsätzlich unterschiedliche Betriebsarten.

Live Capture

Netzwerkpakete werden unmittelbar von einem Netzwerkinterface aufgezeichnet.

Offline-Analyse

Eine bereits vorhandene Capture-Datei wird geöffnet und analysiert.

Für die Offline-Analyse ist keine aktive Netzwerkverbindung erforderlich.

Damit können zuvor gesicherte Netzwerkaufzeichnungen auch später innerhalb von JUMBO BLOCK Rescue untersucht werden.

Capture Interfaces

Für eine Live-Aufzeichnung muss festgelegt werden, von welchem Netzwerkinterface Daten erfasst werden sollen.

Abhängig vom System können beispielsweise vorhanden sein:

Ethernet-Interfaces,

WLAN-Interfaces,

Loopback,

virtuelle Interfaces,

VPN-Interfaces

und weitere vom Betriebssystem bereitgestellte Capture-Schnittstellen.

Vor Beginn einer Aufzeichnung sollte geprüft werden, welches Interface tatsächlich den interessierenden Netzwerkverkehr führt.

Mehrere Interfaces

Wireshark kann gleichzeitig von mehreren Netzwerkinterfaces aufzeichnen.

Dies kann beispielsweise sinnvoll sein, wenn ein System gleichzeitig über Ethernet und eine weitere Netzwerkschnittstelle kommuniziert.

Mehrere Interfaces erhöhen jedoch auch die Menge der aufgezeichneten Daten.

Für eine gezielte Diagnose sollte deshalb nur das tatsächlich benötigte Interface beziehungsweise die benötigte Interface-Gruppe ausgewählt werden.

Capture-Rechte

Das Erfassen von Netzwerkpaketen erfordert unter Linux entsprechende Berechtigungen.

Wireshark trennt aus Sicherheitsgründen die eigentliche Paketerfassung von der umfangreichen grafischen Analyse.

Für die Erfassung wird insbesondere die Komponente:

`dumpcap`

verwendet.

Die konkrete Rechtekonfiguration ist distributions- und installationsabhängig.

Aus der bloßen Paketliste der Baseline darf daher nicht abgeleitet werden, welche Benutzergruppen oder individuellen Capture-Berechtigungen für JUMBO BLOCK Rescue vorkonfiguriert wurden.

Dumpcap

Dumpcap ist Bestandteil der Wireshark-Werkzeugfamilie und für die eigentliche Aufzeichnung von Netzwerkverkehr vorgesehen.

Es kann Netzwerkpakete von Live-Interfaces erfassen und in Capture-Dateien schreiben.

Das Standard-Ausgabeformat von Dumpcap ist PCAPNG.

Wireshark verwendet die Capture-Infrastruktur, um Pakete sicherer von der eigentlichen grafischen Protokollanalyse zu trennen.

Promiscuous Mode

Bei Ethernet-Aufzeichnungen kann ein Interface im Promiscuous Mode betrieben werden.

Dabei wird das Interface so konfiguriert, dass es nicht ausschließlich Frames akzeptiert, die direkt an seine eigene MAC-Adresse adressiert sind.

Dies bedeutet jedoch nicht automatisch, dass dadurch sämtlicher Verkehr eines geswitchten Netzwerkes sichtbar wird.

Grenzen in geswitchten Netzen

In einem modernen Ethernet-Switch wird Unicast-Verkehr normalerweise nur an den Port weitergeleitet, an dem sich das jeweilige Ziel befindet.

Ein Rechner mit Wireshark sieht deshalb typischerweise:

eigenen Verkehr,

Broadcasts,

relevante Multicasts

und Verkehr, den die Netzwerkinfrastruktur tatsächlich an seinen Port weiterleitet.

Um fremden Unicast-Verkehr gezielt zu analysieren, kann eine entsprechend konfigurierte Netzwerkinfrastruktur erforderlich sein, beispielsweise ein Mirror-/SPAN-Port oder ein anderer autorisierter Messpunkt.

Promiscuous Mode allein umgeht diese Netzwerkarchitektur nicht.

WLAN und Monitor Mode

Bei unterstützter WLAN-Hardware und geeigneten Treibern kann für bestimmte Analysen ein Monitor Mode relevant sein.

Dabei kann WLAN-Verkehr auf einer anderen Ebene aufgezeichnet werden als bei einer normalen Verbindung mit einem Access Point.

Ob dies funktioniert, hängt von:

WLAN-Hardware,

Treiber,

Betriebssystem,

Interface-Konfiguration

und gewähltem Kanal

ab.

Die Installation von Wireshark allein garantiert daher keinen funktionierenden WLAN-Monitor-Mode.

Start einer Aufzeichnung

Eine Live-Aufzeichnung wird in der grafischen Oberfläche über das gewünschte Interface gestartet.

Während der Aufzeichnung zeigt Wireshark die eintreffenden Pakete fortlaufend an und dekodiert erkannte Protokolle.

Die Aufzeichnung kann anschließend gestoppt und gespeichert werden.

Die drei zentralen Ansichtsbereiche

Die klassische Wireshark-Ansicht besteht im Wesentlichen aus:

Packet List

Liste der aufgezeichneten Pakete.

Packet Details

Dekodierte Protokollstruktur des ausgewählten Pakets.

Packet Bytes

Tatsächliche Bytewerte des Pakets.

Damit kann ein Paket gleichzeitig auf drei Ebenen untersucht werden:

Übersicht,

Protokollstruktur,

Rohdaten.

Packet List

Die Paketliste zeigt pro Paket typische Informationen wie:

Paketnummer,

Zeit,

Quelle,

Ziel,

Protokoll,

Länge

und eine zusammengefasste Beschreibung.

Die tatsächlich angezeigten Spalten können angepasst werden.

Packet Details

Im Bereich Packet Details zerlegt Wireshark das ausgewählte Paket entsprechend den erkannten Protokollen.

Bei einem Ethernet/IP/TCP-Paket können beispielsweise verschiedene Ebenen getrennt dargestellt werden:

Ethernet,

Internet Protocol,

Transmission Control Protocol,

Anwendungsprotokoll.

Die einzelnen Ebenen können aufgeklappt und detailliert untersucht werden.

Packet Bytes

Packet Bytes zeigt die tatsächlichen Bytes des ausgewählten Pakets.

Damit kann nachvollzogen werden, auf welchen Rohdaten die Dekodierung basiert.

Diese Ansicht ist insbesondere bei:

unbekannten Protokollfeldern,

fehlerhaften Paketen,

proprietären Daten

oder detaillierter Protokollanalyse

relevant.

Protokoll-Dissektoren

Wireshark verwendet sogenannte Dissektoren, um Netzwerkprotokolle zu erkennen und deren Felder zu interpretieren.

Ein Dissektor kennt die Struktur eines bestimmten Protokolls und kann dessen Felder in einer verständlichen Form darstellen.

Wireshark unterstützt eine sehr große Zahl solcher Protokolldissektoren.

Portnummer ist nicht gleich Protokoll

Wireshark verwendet für die Protokollerkennung verschiedene Informationen.

Eine typische Portnummer ist zwar ein wichtiger Hinweis, aber ein Dienst kann auf einem ungewöhnlichen Port betrieben werden.

In bestimmten Fällen kann deshalb eine manuelle Zuordnung erforderlich sein.

Decode As

Mit **Decode As** kann Wireshark angewiesen werden, bestimmten Netzwerkverkehr als ein ausgewähltes Protokoll zu interpretieren.

Das ist beispielsweise nützlich, wenn ein bekannter Dienst auf einem unüblichen Port betrieben wird und deshalb nicht automatisch korrekt erkannt wird.

Decode As verändert nicht die aufgezeichneten Netzwerkpakete.

Es verändert deren Interpretation innerhalb von Wireshark.

Capture Filter und Display Filter

Eine der wichtigsten Grundlagen bei Wireshark ist die klare Trennung zwischen:

Capture Filter

und

Display Filter.

Es handelt sich um zwei unterschiedliche Filtersysteme mit unterschiedlicher Syntax und unterschiedlicher Wirkung.

Sie dürfen nicht miteinander verwechselt werden.

Capture Filter

Ein Capture Filter wird **vor beziehungsweise während der Paketerfassung** angewendet.

Er entscheidet, welche Pakete überhaupt in die Capture-Datei aufgenommen werden.

Pakete, die durch den Capture Filter ausgeschlossen werden, befinden sich anschließend nicht im Mitschnitt.

Sie können daher später nicht durch Änderung eines Display Filters wieder sichtbar gemacht werden.

Capture-Filter-Syntax

Capture Filter verwenden die Filtersyntax der zugrunde liegenden Packet-Capture-Bibliothek.

Typische Filterelemente beziehen sich beispielsweise auf:

Hosts,

Netze,

Ports

und Protokolle.

Beispiele:

```
host 192.0.2.10
```

```
tcp port 443
```

```
udp port 53
```

```
net 192.0.2.0/24
```

Die verwendeten Adressen dienen hier ausschließlich als Dokumentationsbeispiele.

Capture Filter sparsam einsetzen

Capture Filter reduzieren:

Datenmenge,

Speicherbedarf

und Analyseaufwand.

Sie bergen jedoch ein Risiko:

Ein zu eng definierter Capture Filter kann genau die Pakete verwerfen, die später für die Fehleranalyse benötigt werden.

Wenn ausreichend Speicherplatz vorhanden ist, kann es deshalb bei unklaren Fehlerbildern sinnvoll sein, zunächst breiter aufzuzeichnen und anschließend mit Display Filtern einzugrenzen.

Display Filter

Display Filter werden auf bereits erfasste beziehungsweise geöffnete Pakete angewendet.

Sie entscheiden nur, welche Pakete Wireshark momentan anzeigt.

Nicht passende Pakete werden dadurch nicht aus der Capture-Datei gelöscht.

Der Filter kann jederzeit geändert oder entfernt werden.

Beispiel `tcp`

Der Display Filter:

```
tcp
```

zeigt Pakete an, in denen Wireshark TCP entsprechend dekodiert hat.

Beispiel `dns`

Der Display Filter:

```
dns
```

zeigt Pakete an, die Wireshark als DNS dekodiert.

Beispiel IP-Adresse

Ein Filter kann sich auf konkrete Protokollfelder beziehen.

Beispielsweise:

```
ip.addr == 192.0.2.10
```

Damit können IPv4-Pakete gefiltert werden, bei denen die angegebene Adresse entsprechend dem Feldvergleich vorkommt.

Beispiel TCP-Port

Beispiel:

```
tcp.port == 443
```

Damit kann TCP-Verkehr mit dem betreffenden Port angezeigt werden.

Quell- und Zielrichtung

Display Filter können Quelle und Ziel getrennt berücksichtigen.

Beispielsweise:

```
ip.src == 192.0.2.10
```

oder:

```
ip.dst == 192.0.2.10
```

Damit lässt sich die Kommunikationsrichtung gezielt untersuchen.

Logische Verknüpfungen

Display-Filterbedingungen können logisch miteinander kombiniert werden.

Beispielsweise:

```
tcp && ip.addr == 192.0.2.10
```

Damit werden TCP-Pakete angezeigt, die zusätzlich zur angegebenen IP-Bedingung passen.

Komplexe Display Filter ermöglichen eine sehr präzise Eingrenzung großer Paketmitschnitte.

Filter verändern die Daten nicht

Ein Display Filter löscht keine Pakete.

Er verändert ausschließlich die aktuelle Darstellung.

Das ist für die Analyse entscheidend:

Captured beschreibt die tatsächlich aufgezeichneten Pakete.

Displayed beschreibt die aufgrund des aktuellen Display Filters angezeigten Pakete.

Paket suchen

Wireshark bietet Funktionen zum gezielten Auffinden von Paketen.

Gesucht werden kann abhängig von der gewählten Funktion beispielsweise nach:

Display-Filter-Kriterien,

Hexwerten,

Zeichenketten

oder anderen Paketmerkmalen.

Dies erleichtert die Untersuchung großer Capture-Dateien.

Pakete markieren

Interessante Pakete können markiert werden.

Damit lassen sich relevante Stellen innerhalb einer umfangreichen Aufzeichnung schneller wiederfinden.

Markierungen verändern nicht den eigentlichen Netzwerkverkehr.

Zeitdarstellung

Wireshark kann Paketzeiten auf unterschiedliche Weise darstellen.

Für die Analyse können beispielsweise relevant sein:

absolute Uhrzeit,

Zeit seit Beginn der Aufzeichnung,

Zeit seit dem vorherigen Paket

oder andere Zeitbezüge.

Die richtige Zeitdarstellung hängt vom Fehlerbild ab.

Zeitstempel

Die Zeitstempel eines Captures stammen aus der jeweiligen Capture-Umgebung.

Ihre Genauigkeit hängt unter anderem von:

Betriebssystem,

Capture-Mechanismus,

Hardware,

Treiber

und Zeitkonfiguration

ab.

Bei Vergleichen mehrerer Systeme muss deshalb berücksichtigt werden, ob deren Uhren ausreichend synchronisiert waren.

TCP-Verbindungsaufbau

Wireshark eignet sich hervorragend zur Untersuchung eines TCP-Verbindungsaufbaus.

Der klassische Ablauf besteht aus:

SYN

SYN/ACK

ACK.

Fehlt eine erwartete Antwort, kann die Paketaufzeichnung wichtige Hinweise darauf geben, an welcher Stelle die Kommunikation scheitert.

TCP-Verbindungsabbau

Auch FIN- und RST-Vorgänge können untersucht werden.

Damit kann beispielsweise festgestellt werden, ob eine Verbindung:

geordnet beendet,

von einer Seite zurückgesetzt

oder unerwartet abgebrochen

wurde.

TCP Retransmissions

Wireshark kann verschiedene TCP-Analysehinweise anzeigen.

Dazu können beispielsweise erkannte Retransmissions gehören.

Solche Hinweise müssen jedoch im Kontext bewertet werden.

Eine als Retransmission klassifizierte Situation bedeutet nicht automatisch, dass die Ursache eindeutig im Netzwerk liegt.

Capture-Verluste, asymmetrische Aufzeichnung oder andere Effekte können die Analyse beeinflussen.

Checksums

Wireshark kann Prüfsummen verschiedener Protokolle untersuchen.

Bei lokalen Captures können jedoch scheinbar fehlerhafte Prüfsummen auftreten, obwohl die tatsächlich übertragene Netzwerkkommunikation korrekt ist.

Eine wichtige Ursache ist **Checksum Offloading**.

Dabei berechnet die Netzwerkkarte bestimmte Prüfsummen erst nach dem Punkt, an dem Wireshark das Paket im Host erfasst hat.

Ein angezeigter Prüfsummenfehler ist deshalb nicht automatisch ein real auf dem Netzwerk übertragener Fehler.

DNS-Analyse

DNS-Verkehr kann mit:

`dns`

gefiltert werden.

Damit lassen sich beispielsweise:

DNS-Abfragen,

Antworten,

angefragte Namen,

Antwortadressen,

Fehlercodes

und Antwortzeiten

untersuchen.

Für gezielte aktive DNS-Abfragen steht zusätzlich A.22 Dig zur Verfügung.

DHCP-Analyse

Wireshark kann DHCP-Kommunikation detailliert darstellen.

Damit lässt sich beispielsweise untersuchen, ob ein Client:

eine Adresse anfordert,

einen DHCP-Server erreicht,

ein Angebot erhält

und eine Konfiguration zugewiesen bekommt.

Dies ist bei Systemen ohne erwartete Netzwerkkonfiguration besonders hilfreich.

ARP-Analyse

Mit:

`arp`

kann ARP-Verkehr angezeigt werden.

Damit lassen sich lokale IPv4-Adressauflösungen auf Ethernet-Ebene untersuchen.

Dies ergänzt A.20 ARP Scan:

ARP Scan erzeugt gezielte ARP-Anfragen zur Geräteerkennung.

Wireshark kann die tatsächlich übertragenen ARP-Pakete sichtbar machen.

ICMP-Analyse

Mit:

`icmp`

kann IPv4-ICMP-Verkehr untersucht werden.

Damit können beispielsweise Echo Requests, Echo Replies und bestimmte Fehlermeldungen sichtbar gemacht werden.

Für IPv6 existieren entsprechende ICMPv6-Protokollfelder.

HTTP

Unverschlüsselter HTTP-Verkehr kann – sofern er im Capture sichtbar ist – detailliert dekodiert werden.

Dabei können unter anderem Requests, Responses, Header und weitere Protokollinformationen untersucht werden.

Das bedeutet gleichzeitig:

Unverschlüsselte Netzwerkprotokolle können sensible Informationen offen übertragen.

Capture-Dateien sind daher entsprechend zu schützen.

TLS

Bei TLS-verschlüsselter Kommunikation kann Wireshark die sichtbaren TLS-Protokollinformationen analysieren.

Die verschlüsselten Anwendungsdaten sind jedoch nicht allein deshalb im Klartext sichtbar, weil Wireshark den Datenverkehr aufgezeichnet hat.

Für eine Entschlüsselung sind geeignete Schlüssel beziehungsweise Session-Secrets und passende technische Voraussetzungen erforderlich.

Verschlüsselung bleibt Verschlüsselung

Wireshark „knackt“ eine TLS-Verbindung nicht automatisch.

Ohne die erforderlichen kryptografischen Informationen bleibt der verschlüsselte Nutzdateninhalt geschützt.

Trotzdem bleiben Metadaten der Kommunikation sichtbar, beispielsweise:

IP-Adressen,

Ports,

Paketgrößen,

Zeitpunkte

und bestimmte unverschlüsselte Protokollinformationen.

Follow Stream

Wireshark kann zusammengehörige Kommunikationsströme rekonstruieren und in einer zusammenhängenden Ansicht darstellen.

Über:

Analyze → Follow

können unterstützte Streams verfolgt werden.

Dazu gehören unter anderem TCP, UDP, TLS, HTTP, HTTP/2, QUIC, WebSocket und weitere von Wireshark unterstützte Streamtypen.

Follow TCP Stream

Follow TCP Stream stellt die Nutzdaten einer ausgewählten TCP-Verbindung in ihrer Kommunikationsreihenfolge dar.

Damit lässt sich wesentlich leichter erkennen, welche Daten innerhalb einer Verbindung ausgetauscht wurden, als durch die isolierte Betrachtung einzelner Pakete.

Wireshark setzt dabei gleichzeitig einen passenden Display Filter für den betreffenden Stream.

Follow Stream verändert Capture nicht

Follow Stream rekonstruiert und filtert die Darstellung.

Die ursprünglichen Pakete der Capture-Datei bleiben erhalten.

Die Streamansicht kann deshalb geschlossen und die Analyse anschließend auf Paketebene fortgesetzt werden.

Statistics

Wireshark besitzt umfangreiche Statistikfunktionen.

Sie befinden sich insbesondere im Menü:

Statistics

Damit können große Capture-Dateien zusammenfassend untersucht werden.

Protocol Hierarchy

Die Protocol Hierarchy zeigt, welche Protokolle innerhalb einer Aufzeichnung erkannt wurden und welchen Anteil sie am erfassten Verkehr besitzen.

Damit lässt sich schnell feststellen, welche Protokolle einen Capture dominieren.

Endpoints

Die Endpoint-Statistik fasst einzelne Kommunikationsendpunkte zusammen.

Je nach Protokollebene können beispielsweise:

MAC-Adressen,

IPv4-Adressen,

IPv6-Adressen

und andere Endpunkte

dargestellt werden.

Dies ist hilfreich, um besonders aktive Systeme innerhalb eines Captures zu identifizieren.

Conversations

Die Conversations-Ansicht fasst Kommunikationsbeziehungen zwischen jeweils zwei Endpunkten zusammen.

Sie zeigt statistische Werte für einzelne Verbindungen beziehungsweise Kommunikationspaare.

Damit kann beispielsweise erkannt werden:

welche Systeme miteinander kommunizieren,

wie viele Pakete übertragen wurden,

welche Datenmengen entstanden,

und wann eine Kommunikation begann.

I/O Graphs

Wireshark kann Netzwerkaktivität über die Zeit grafisch darstellen.

I/O Graphs sind hilfreich, um beispielsweise:

Verkehrsspitzen,

periodische Kommunikation,

Unterbrechungen

oder zeitliche Zusammenhänge

sichtbar zu machen.

Expert Information

Wireshark stellt sogenannte Expert Information bereit.

Diese Funktion fasst bestimmte auffällige beziehungsweise analyserelevante Protokollereignisse zusammen.

Dazu können Hinweise auf:

Fehler,

Warnungen,

ungewöhnliche Abläufe

und andere Protokollereignisse

gehören.

Expert Information ist kein endgültiges Urteil

Expert Information erleichtert das Auffinden interessanter Ereignisse.

Die Einträge sind jedoch Analysehinweise.

Sie beweisen nicht automatisch einen Netzwerkdefekt oder Sicherheitsvorfall.

Jeder Fund muss im Kontext des betreffenden Protokolls und Captures bewertet werden.

Capture-Dateien

Wireshark kann Netzwerkaufzeichnungen speichern und später erneut öffnen.

Zentrale Formate sind insbesondere:

PCAP

und

PCAPNG.

PCAPNG

PCAPNG ist das moderne Standardformat der Wireshark-Capture-Werkzeugfamilie.

Es kann gegenüber klassischem PCAP zusätzliche Metadaten speichern.

Dazu können beispielsweise Informationen über Capture-Interfaces und Kommentare gehören.

Für mehrere gleichzeitig aufgezeichnete Interfaces ist PCAPNG besonders relevant.

PCAP

PCAP ist ein älteres und sehr weit verbreitetes Capture-Format.

Es wird von zahlreichen Netzwerkwerkzeugen unterstützt.

Damit eignet es sich gut für den Austausch mit anderen Analyseprogrammen, bietet jedoch nicht alle Metadatenmöglichkeiten von PCAPNG.

Capture-Dateien enthalten sensible Daten

Ein Capture kann deutlich mehr Informationen enthalten als ein gewöhnliches Diagnoseprotokoll.

Abhängig vom Netzwerkverkehr können darin enthalten sein:

interne IP-Adressen,

Hostnamen,

DNS-Anfragen,

Kommunikationspartner,

Protokolldaten,

unverschlüsselte Inhalte,

Authentifizierungsinformationen bei unsicheren Protokollen

und andere sensible Betriebsdaten.

Capture-Dateien müssen deshalb entsprechend geschützt werden.

Capture-Dateien nicht unkontrolliert weitergeben

Vor der Weitergabe eines Captures muss geprüft werden, welche Informationen darin enthalten sind.

Das gilt insbesondere für:

Unternehmensnetze,

KRITIS-Umgebungen,

Produktionssysteme,

personenbezogene Kommunikation

und sicherheitsrelevante Infrastrukturen.

Eine PCAP-Datei sollte nicht allein deshalb als unkritisch betrachtet werden, weil sie „nur Netzwerkpakete“ enthält.

Capture File Properties

Wireshark kann Eigenschaften einer Capture-Datei anzeigen.

Dazu gehören unter anderem:

Dateipfad,

Dateigröße,

kryptografische Hashwerte,

Dateiformat,

Encapsulation,

Zeitpunkt des ersten und letzten Pakets,

Capture-Informationen,

Interfaces

und statistische Angaben.

Dies ist für die technische Dokumentation eines Mitschnitts hilfreich.

Capture-Kommentare

PCAPNG kann Kommentare zur gesamten Capture-Datei speichern.

Damit können beispielsweise Analysekontext oder technische Hinweise dokumentiert werden.

Wird eine Capture-Datei beweisrelevant verwendet, müssen Änderungen an Metadaten beziehungsweise Kommentaren entsprechend berücksichtigt und dokumentiert werden.

Ring Buffer

Wireshark beziehungsweise seine Capture Engine kann bei Langzeitaufzeichnungen mehrere Capture-Dateien verwenden und diese rotieren.

Ein Ring Buffer verhindert, dass eine einzelne Capture-Datei unbegrenzt wächst.

Abhängig von der Konfiguration können ältere Capture-Dateien überschrieben werden, wenn eine festgelegte Anzahl erreicht ist.

Dies muss bei Langzeitdiagnosen bewusst konfiguriert werden.

Automatisches Stoppen

Eine Aufzeichnung kann anhand verschiedener Bedingungen automatisch beendet werden.

Dazu können gehören:

erfasste Datenmenge,

verstrichene Zeit,

Anzahl von Paketen

oder entsprechende Capture-Bedingungen.

Damit lässt sich verhindern, dass eine unbeaufsichtigte Aufzeichnung unbegrenzt weiterläuft.

Paketverluste beim Capture

Wireshark kann nur Pakete analysieren, die tatsächlich erfasst wurden.

Bei hoher Netzlast, unzureichender Systemleistung oder ungeeigneter Capture-Konfiguration können Pakete während der Aufzeichnung verloren gehen.

Fehlende Pakete können anschließend die Protokollanalyse verfälschen.

Bei kritischen Untersuchungen muss deshalb auch die Qualität des Captures selbst bewertet werden.

Snap Length

Die sogenannte Snapshot Length beziehungsweise Snaplen bestimmt, wie viele Bytes eines Pakets aufgezeichnet werden.

Wird nur ein Teil jedes Pakets gespeichert, reduziert dies Speicherbedarf und Datenmenge.

Gleichzeitig können für eine spätere Analyse benötigte Nutzdaten fehlen.

Für eine unbekannte Fehlerursache sollte eine verkürzte Snaplen daher nur bewusst eingesetzt werden.

Namensauflösung

Wireshark kann numerische Adressen und Ports teilweise in Namen auflösen.

Dies erleichtert die Lesbarkeit.

Eine Namensauflösung kann jedoch zusätzliche Netzwerkkommunikation erzeugen und die Interpretation beeinflussen.

Bei einer präzisen Analyse sollte deshalb immer erkennbar bleiben, welche tatsächliche numerische Adresse beziehungsweise Portnummer hinter einem angezeigten Namen steht.

Export von Paketen

Wireshark kann ausgewählte beziehungsweise gefilterte Pakete in neue Capture-Dateien exportieren.

Damit lässt sich aus einem großen Mitschnitt ein kleinerer relevanter Ausschnitt erzeugen.

Das Original sollte dabei unverändert erhalten bleiben, wenn eine spätere vollständige Analyse erforderlich sein könnte.

Export von Packet Bytes

Ausgewählte Paketbytes können ebenfalls exportiert werden.

Dies ist beispielsweise interessant, wenn ein bestimmter binärer Inhalt anschließend mit anderen Analysewerkzeugen untersucht werden soll.

Export Objects

Für bestimmte unterstützte Protokolle kann Wireshark über **Export Objects** übertragene Objekte extrahieren.

Solche extrahierten Dateien stammen unmittelbar aus dem beobachteten Netzwerkverkehr.

Sie sind daher grundsätzlich als nicht vertrauenswürdig zu behandeln.

Eine extrahierte Datei sollte nicht allein deshalb ausgeführt werden, weil Wireshark sie erfolgreich rekonstruiert hat.

Capinfos

Zur Wireshark-Werkzeugfamilie gehört:

`capinfos`.

Capinfos liest Capture-Dateien und gibt Informationen und Statistiken darüber aus.

Dies kann beispielsweise für eine schnelle terminalbasierte Bestandsaufnahme einer Capture-Datei verwendet werden.

Editcap

`editcap` kann Capture-Dateien lesen, ausgewählte Pakete verarbeiten und eine neue Capture-Datei erzeugen.

Zu seinen Aufgaben gehören unter anderem das Ausschneiden beziehungsweise Auswählen von Capture-Bereichen und die Konvertierung von Capture-Formaten.

Das Original sollte bei wichtigen Untersuchungen erhalten bleiben.

Mergecap

`mergecap` kann mehrere Capture-Dateien zu einer gemeinsamen Datei zusammenführen.

Standardmäßig werden die Pakete anhand ihrer Zeitstempel chronologisch zusammengeführt.

Dies ist beispielsweise nützlich, wenn mehrere Teilaufzeichnungen gemeinsam analysiert werden sollen.

Wireshark und TCPDump

A.19 TCPDump erfüllt einen Teil derselben Grundaufgabe wie Wireshark:

Netzwerkpakete erfassen und untersuchen.

Der Schwerpunkt unterscheidet sich jedoch.

Wireshark

bietet eine umfangreiche grafische Protokollanalyse.

TCPDump

ist auf schnelle terminalbasierte Paketaufzeichnung und Filterung ausgerichtet.

Eine mit TCPDump erzeugte PCAP-Datei kann anschließend mit Wireshark detailliert untersucht werden.

Damit ergänzen sich beide Programme unmittelbar.

Wireshark und Nmap

Nmap erzeugt aktiv Netzwerkverkehr.

Wireshark kann diesen Verkehr beobachten.

Ein kombinierter Diagnoseablauf kann deshalb lauten:

Wireshark-Aufzeichnung starten.

Nmap-Scan durchführen.

Aufzeichnung stoppen.

Nmap-Probes und Antworten anschließend in Wireshark untersuchen.

Damit kann nachvollzogen werden, warum Nmap beispielsweise einen Port als `open`, `closed` oder `filtered` klassifiziert hat.

Wireshark und GUFW

GUFW/UFW kontrolliert Netzwerkverkehr anhand von Firewall-Regeln.

Wireshark analysiert Pakete, die am jeweiligen Capture-Punkt sichtbar sind.

Damit erfüllen beide Werkzeuge unterschiedliche Aufgaben:

GUFW/UFW → Netzwerkzugriff kontrollieren

Wireshark → Netzwerkkommunikation beobachten und analysieren

Bei Firewall-Problemen können beide Perspektiven gemeinsam hilfreich sein.

Wireshark und Dig

Dig erzeugt gezielte DNS-Anfragen und zeigt deren DNS-Antworten strukturiert an.

Wireshark kann gleichzeitig den tatsächlichen DNS-Paketverkehr beobachten.

Damit können beispielsweise Unterschiede zwischen:

Anwendungsergebnis

und

tatsächlichem Netzwerkverkehr

untersucht werden.

Wireshark und MTR

MTR erzeugt Diagnoseverkehr zur Untersuchung des Netzwerkpfads.

Wireshark kann diesen Verkehr auf Paketebene sichtbar machen.

MTR liefert dabei die übersichtliche Pfad- und Latenzperspektive, Wireshark die detaillierte Paketperspektive.

Typischer Rescue-Ablauf

Ein kontrollierter Wireshark-Einsatz kann folgendermaßen erfolgen:

1. Problem definieren

Beispielsweise:

DNS funktioniert nicht,

Verbindung wird zurückgesetzt,

Dienst reagiert nicht,

DHCP liefert keine Adresse.

2. Richtiges Interface bestimmen

Das Interface auswählen, über das die betreffende Kommunikation tatsächlich läuft.

3. Capture Filter nur bei Bedarf

Bei ausreichendem Speicher und unklarer Ursache zunächst nicht unnötig stark einschränken.

4. Aufzeichnung starten

5. Problem reproduzieren

Nur die für die Diagnose benötigte Aktion ausführen.

6. Aufzeichnung stoppen

Dadurch bleibt der Capture überschaubar.

7. Display Filter anwenden

Beispielsweise:

```
dns
```

```
tcp
```

```
arp
```

oder ein gezielter Host-/Portfilter.

8. Relevante Pakete untersuchen

Packet List, Packet Details und Packet Bytes gemeinsam betrachten.

9. Streams und Statistiken verwenden

Falls erforderlich Follow Stream, Conversations, Endpoints oder Expert Information einsetzen.

10. Capture speichern

Bei dokumentationsrelevanten Einsätzen Original-Capture unverändert aufbewahren und weitere Bearbeitungen an einer Kopie durchführen.

Beispiel: DNS-Problem

Eine Aufzeichnung wird gestartet.

Anschließend wird die problematische Namensauflösung ausgelöst.

Danach wird der Capture gestoppt.

Display Filter:

```
dns
```

Nun kann geprüft werden:

Wurde überhaupt eine DNS-Anfrage gesendet?

An welchen Server?

Kam eine Antwort zurück?

Welcher Response Code wurde geliefert?

Wie lange dauerte die Antwort?

Damit lässt sich unterscheiden, ob das Problem bereits vor der DNS-Anfrage, auf dem Transportweg oder innerhalb der DNS-Antwort liegt.

Beispiel: TCP-Dienst nicht erreichbar

Display Filter:

```
tcp.port == PORT
```

Anschließend kann untersucht werden:

Wird ein SYN gesendet?

Kommt ein SYN/ACK zurück?

Kommt ein RST?

Bleibt eine Antwort vollständig aus?

Damit kann ein abstraktes „Verbindung funktioniert nicht“ auf konkrete Paketereignisse zurückgeführt werden.

Beispiel: DHCP-Problem

Die DHCP-Kommunikation wird während des Verbindungsaufbaus aufgezeichnet.

Anschließend kann geprüft werden, welche Schritte des DHCP-Ablaufs tatsächlich sichtbar sind.

Dadurch lässt sich beispielsweise unterscheiden zwischen:

Client sendet keine Anfrage,

Server antwortet nicht,

Antwort erreicht Client nicht,

oder Konfiguration wird aus einem anderen Grund nicht übernommen.

Grenzen

Wireshark sieht nur Pakete, die am gewählten Capture-Punkt tatsächlich verfügbar sind.

Promiscuous Mode macht nicht automatisch sämtlichen Verkehr eines geswitchten Netzwerks sichtbar.

Verschlüsselte Nutzdaten bleiben ohne geeignete kryptografische Informationen verschlüsselt.

Capture-Verluste können Analysen verfälschen.

Checksum Offloading kann scheinbare Prüfsummenfehler erzeugen.

Ein Display Filter zeigt nur eine Teilmenge der vorhandenen Pakete.

Ein Capture Filter kann Pakete dauerhaft aus der Aufzeichnung ausschließen.

Protokolldissektoren können unbekannte oder ungewöhnliche Protokolle nicht immer korrekt interpretieren.

Wireshark ist kein automatisches Intrusion-Detection-System und kein vollständiger Malware-Scanner.

Sicherheitshinweise

Netzwerkverkehr nur in Umgebungen aufzeichnen, für die eine entsprechende Berechtigung besteht.

Capture-Dateien können hochsensible Informationen enthalten.

Unverschlüsselte Protokolle können Zugangsdaten oder andere vertrauliche Inhalte offen übertragen.

Capture-Dateien entsprechend geschützt speichern.

Capture Filter und Display Filter nicht verwechseln.

Bei Langzeitaufzeichnungen Speicherbedarf und Ring-Buffer-Konfiguration berücksichtigen.

Bei forensisch relevanten Captures das Original unverändert erhalten.

Extrahierte Netzwerkobjekte grundsätzlich als nicht vertrauenswürdig behandeln.

Scheinbare Checksum-Fehler nicht ohne Prüfung von Offloading-Effekten als reale Netzwerkfehler interpretieren.

Eine Paketaufzeichnung zeigt die Perspektive des gewählten Capture-Punkts und nicht zwangsläufig den vollständigen Kommunikationsweg.

Technische Einordnung in JUMBO BLOCK Rescue

Wireshark bildet innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue die zentrale **grafische Paket- und Protokollanalyse**.

Während:

A.17 Nmap

aktiv Hosts, Ports und Dienste untersucht,

beobachtet:

A.18 Wireshark

die tatsächlich am Capture-Punkt sichtbaren Netzwerkpakete und zerlegt sie bis auf Protokoll- und Byteebene.

Seine wesentlichen Analyseebenen sind:

Capture → Netzwerkverkehr erfassen

Packet List → Kommunikationsablauf überblicken

Packet Details → Protokollfelder untersuchen

Packet Bytes → tatsächliche Rohdaten kontrollieren

Display Filter → relevante Kommunikation isolieren

Follow Stream → zusammengehörige Kommunikation rekonstruieren

Statistics → größere Datenmengen zusammenfassen

Damit entsteht zusammen mit Nmap eine besonders leistungsfähige Kombination:

Nmap erzeugt eine kontrollierte aktive Untersuchung.

Wireshark zeigt, was dabei tatsächlich über das Netzwerk kommuniziert wird.

Mit dem folgenden:

A.19 TCPDump

kommt dieselbe grundlegende Paketperspektive zusätzlich als leichtgewichtige terminalbasierte Lösung hinzu.

Wireshark übernimmt damit in JUMBO BLOCK Rescue die Rolle des detaillierten grafischen Netzwerkprotokollanalysators für Live-Diagnose und gespeicherte Capture-Dateien.

A.19 TCPDump

Bereich: Netzwerk
Programm: TCPDump
Baseline 20261001: TCPDump 4.99.5
Installiertes Paket: `tcpdump 4.99.5-2`
Bedienung: Kommandozeile
Aufgabe: Terminalbasierte Aufzeichnung, Filterung und Analyse von Netzwerkpaketen

Zweck

TCPDump ist ein kommandozeilenbasiertes Werkzeug zur Aufzeichnung und Untersuchung von Netzwerkverkehr.

Es kann Pakete unmittelbar von Netzwerkinterfaces erfassen, anhand definierter Filter auswählen, Informationen über die Pakete im Terminal ausgeben und Netzwerkverkehr in Capture-Dateien speichern.

Diese Capture-Dateien können anschließend erneut mit TCPDump oder mit anderen Analyseprogrammen wie Wireshark untersucht werden.

Innerhalb von JUMBO BLOCK Rescue stellt TCPDump damit die kompakte terminalbasierte Ergänzung zu Wireshark dar.

Einsatzgebiet im Rescue-System

TCPDump eignet sich insbesondere für:

schnelle Netzwerkdiagnosen im Terminal,

Aufzeichnung von Netzwerkverkehr ohne grafische Oberfläche,

gezielte Untersuchung einzelner Hosts oder Ports,

DNS-, ARP-, ICMP-, TCP- und UDP-Diagnosen,

Erstellung von PCAP-Dateien,

Langzeit- beziehungsweise rotierende Aufzeichnungen,

Analyse bereits vorhandener Capture-Dateien,

Kontrolle von Nmap-, MTR- oder Dig-Verkehr,

und Vorbereitung eines Captures für die spätere detaillierte Analyse mit Wireshark.

Grundlegender Aufruf

Ein grundlegender Aufruf lautet:

```
tcpdump
```

TCPDump versucht damit, Pakete auf einem geeigneten Netzwerkinterface zu erfassen.

Für einen kontrollierten Rescue-Einsatz sollte das gewünschte Interface jedoch möglichst ausdrücklich festgelegt werden.

Netzwerkinterfaces anzeigen

Mit:

```
tcpdump -D
```

werden die für TCPDump verfügbaren Capture-Interfaces aufgelistet.

Dies sollte bei unbekannter Interface-Konfiguration einer der ersten Schritte sein.

Abhängig vom System können beispielsweise Ethernet-, WLAN-, Loopback-, VPN- und virtuelle Interfaces vorhanden sein.

Interface auswählen

Mit:

```
-i
```

wird das gewünschte Capture-Interface festgelegt.

Beispiel:

```
sudo tcpdump -i INTERFACE
```

Damit wird der Mitschnitt gezielt auf diesem Interface durchgeführt.

Der konkrete Interfacename muss aus der tatsächlichen Systemkonfiguration übernommen werden.

any

Unter Linux kann:

```
-i any
```

verwendet werden, um Pakete von mehreren verfügbaren Interfaces über das spezielle `any`-Capture-Interface zu erfassen.

Beispiel:

```
sudo tcpdump -i any
```

Dies ist für eine erste Fehlersuche nützlich, wenn noch nicht bekannt ist, über welches Interface die relevante Kommunikation läuft.

Die Eigenschaften einer Aufzeichnung über `any` unterscheiden sich jedoch von einem Capture auf einem einzelnen physischen Interface.

Capture-Berechtigungen

Die Live-Aufzeichnung von Netzwerkpaketen benötigt entsprechende Systemberechtigungen.

Welche Rechte einem normalen Benutzer innerhalb einer konkreten Installation zur Verfügung stehen, hängt von der Systemkonfiguration ab.

Im Handbuch wird deshalb nicht vorausgesetzt, dass jeder Benutzer ohne erhöhte Rechte einen Live-Capture starten kann.

Beispiele werden bei erforderlichem privilegiertem Zugriff mit `sudo` dargestellt.

Standardausgabe

Ohne `-w` dekodiert TCPDump die erfassten Pakete und gibt eine textuelle Zusammenfassung im Terminal aus.

Die genaue Darstellung hängt unter anderem ab von:

Protokoll,

gewählten Optionen,

Namensauflösung,

Verbose-Level

und Capture-Inhalt.

Paketanzahl begrenzen

Mit:

`-c ANZAHL`

wird die Aufzeichnung beendet, nachdem die angegebene Zahl von Paketen verarbeitet wurde.

Beispiel:

```
sudo tcpdump -i INTERFACE -c 100
```

Dies ist bei kurzen Diagnoseaufgaben besonders nützlich.

Der Capture läuft dadurch nicht unbegrenzt weiter.

Namensauflösung deaktivieren

Mit:

`-n`

werden Adressen nicht in Namen umgewandelt.

Beispiel:

```
sudo tcpdump -n -i INTERFACE
```

Damit werden beispielsweise IP-Adressen numerisch dargestellt.

Dies hat mehrere Vorteile:

Die Ausgabe bleibt eindeutig.

Zusätzliche Namensauflösungen werden vermieden.

Die Analyse wird nicht durch DNS-Aktivität verfälscht beziehungsweise verzögert.

Für technische Rescue-Diagnosen ist `-n` deshalb häufig sinnvoll.

Ports ebenfalls numerisch darstellen

Mit:

`-nn`

wird zusätzlich die Umwandlung von Portnummern in Dienstnamen unterdrückt.

Beispiel:

```
sudo tcpdump -nn -i INTERFACE
```

Damit bleibt beispielsweise:

`443`

als Portnummer sichtbar, statt aufgrund einer lokalen Dienstedatenbank als Dienstname dargestellt zu werden.

Dies ist bei präzisen Netzwerkdiagnosen oft die übersichtlichste Darstellung.

Verbose-Ausgabe

TCPDump unterstützt mehrere Stufen ausführlicherer Ausgabe.

`-v`

liefert zusätzliche Informationen.

`-vv`

erhöht die Ausführlichkeit weiter.

`-vvv`

liefert die maximal vorgesehene Verbose-Stufe dieser Optionsgruppe.

Mehr Ausgabe bedeutet jedoch nicht automatisch eine bessere Diagnose.

Für umfangreiche Captures kann eine zu detaillierte Terminalausgabe schnell unübersichtlich werden.

Zeitstempel

TCPDump stellt Pakete mit Zeitinformationen dar.

Mit der Option:

`-t`

kann die Ausgabe des Zeitstempels unterdrückt werden.

Zusätzliche `-t` verändern die Zeitdarstellung weiter.

Für Fehleranalysen sind Zeitstempel häufig wichtig und sollten deshalb nicht ohne Grund entfernt werden.

Absolute Zeit

Die Zeitdarstellung kann so gewählt werden, dass absolute Datums- und Zeitinformationen ausgegeben werden.

Dies ist insbesondere hilfreich, wenn TCPDump-Ergebnisse mit:

Systemlogs,

Firewall-Logs,

Serverlogs

oder anderen Ereignisprotokollen

korreliert werden sollen.

Filterausdruck

Eine der wichtigsten Eigenschaften von TCPDump ist die Möglichkeit, einen Capture-Filterausdruck direkt auf der Kommandozeile anzugeben.

Grundprinzip:

```
tcpdump [Optionen] AUSDRUCK
```

Der Ausdruck bestimmt, welche Pakete für die Ausgabe beziehungsweise Aufzeichnung ausgewählt werden.

BPF-Filter

TCPDump verwendet die Filterausdruckssprache der Packet-Capture-Infrastruktur, häufig als BPF- beziehungsweise libpcap-Filter bezeichnet.

Diese Filtersprache ist eng mit den **Capture Filters** von Wireshark verwandt.

Sie ist jedoch nicht mit den Wireshark **Display Filters** identisch.

Das ist eine wichtige Unterscheidung.

Host filtern

Beispiel:

```
sudo tcpdump -nn host 192.0.2.10
```

Damit werden Pakete ausgewählt, bei denen die betreffende Adresse entsprechend dem Filter als Host beteiligt ist.

Quellhost

Mit:

`src host`

kann gezielt nach einer Quelle gefiltert werden.

Beispiel:

```
sudo tcpdump -nn src host 192.0.2.10
```

Zielhost

Mit:

`dst host`

wird auf ein Ziel eingeschränkt.

Beispiel:

```
sudo tcpdump -nn dst host 192.0.2.10
```

Netzwerk filtern

Mit:

`net`

kann ein Netzwerkbereich angegeben werden.

Beispiel:

```
sudo tcpdump -nn net 192.0.2.0/24
```

Die Adresse dient ausschließlich als Dokumentationsbeispiel.

Port filtern

Mit:

`port`

kann ein bestimmter Port ausgewählt werden.

Beispiel:

```
sudo tcpdump -nn port 53
```

Der Ausdruck allein unterscheidet noch nicht zwingend zwischen TCP und UDP.

TCP filtern

Mit:

`tcp`

wird TCP-Verkehr ausgewählt.

Beispiel:

```
sudo tcpdump -nn tcp
```

UDP filtern

Mit:

`udp`

wird UDP-Verkehr ausgewählt.

Beispiel:

```
sudo tcpdump -nn udp
```

TCP-Port

Filter können kombiniert werden:

```
sudo tcpdump -nn tcp port 443
```

Damit wird TCP-Verkehr für Port 443 ausgewählt.

UDP-Port

Beispiel:

```
sudo tcpdump -nn udp port 53
```

Damit wird UDP-Verkehr für Port 53 ausgewählt.

DNS-Verkehr

Ein einfacher DNS-bezogener Capture kann beispielsweise lauten:

```
sudo tcpdump -nn port 53
```

Damit können TCP- und UDP-Pakete erfasst werden, die zum Portfilter passen.

Für eine ausschließlich UDP-bezogene Untersuchung:

```
sudo tcpdump -nn udp port 53
```

ARP

ARP-Verkehr kann mit:

```
sudo tcpdump -nn arp
```

untersucht werden.

Dies ist bei lokalen IPv4-Netzwerkproblemen besonders nützlich.

Damit kann beispielsweise geprüft werden, ob:

ARP-Anfragen gesendet

und

ARP-Antworten empfangen

werden.

ICMP

IPv4-ICMP-Verkehr kann mit:

```
sudo tcpdump -nn icmp
```

gefiltert werden.

Damit können beispielsweise Echo Requests, Echo Replies und bestimmte ICMP-Fehlermeldungen untersucht werden.

IPv6

TCPDump unterstützt auch IPv6-Verkehr.

Ein Filter kann beispielsweise mit:

```
ip6
```

auf IPv6 eingeschränkt werden.

ICMPv6 kann entsprechend gezielt gefiltert werden.

Logische Verknüpfungen

Filterbedingungen können mit logischen Operatoren kombiniert werden.

Wesentliche Operatoren sind:

`and`

`or`

`not`.

Beispiel:

```
tcp and port 443
```

AND

Beispiel:

```
host 192.0.2.10 and tcp port 443
```

Beide Bedingungen müssen erfüllt sein.

OR

Beispiel:

```
port 53 or port 123
```

Damit werden Pakete ausgewählt, die zu einer der beiden Bedingungen passen.

NOT

Beispiel:

```
not port 22
```

Damit können Pakete ausgeschlossen werden, die dem betreffenden Filter entsprechen.

Klammern

Bei komplexeren Ausdrücken können Klammern erforderlich sein.

Da die Shell Klammern und andere Sonderzeichen selbst interpretieren kann, sollte ein komplexer Filterausdruck sinnvollerweise in Anführungszeichen gesetzt werden.

Beispielprinzip:

```
sudo tcpdump -nn 'host 192.0.2.10 and (tcp port 80 or tcp port 443)'
```

Damit wird zugleich eindeutig festgelegt, welche logische Gruppierung beabsichtigt ist.

Capture Filter reduzieren die Datenmenge

Ein Filter wird während der Erfassung angewendet.

Pakete, die nicht zum Filter passen, werden bei einem mit `-w` gespeicherten Capture nicht in die entsprechende Datei aufgenommen.

Das reduziert:

Speicherbedarf,

Schreiblast

und spätere Datenmenge.

Es bedeutet aber auch:

Nicht aufgezeichnete Pakete können später nicht zurückgeholt werden.

Bei einer unbekannten Fehlerursache sollte ein Filter daher nicht unnötig eng gewählt werden.

Snaplen

Mit:

```
-s
```

wird die Snapshot Length festgelegt.

Sie bestimmt, wie viele Bytes eines Pakets aufgezeichnet werden.

Bei TCPDump 4.99.5 bedeutet:

```
-s 0
```

die Verwendung der Standard-Maximalgröße von **262144 Bytes**.

Verkürzte Pakete

Eine kleinere Snaplen reduziert Datenmenge und Speicherbedarf.

Sie kann jedoch dazu führen, dass nur der Anfang eines Pakets aufgezeichnet wird.

Für spätere Protokoll- oder Nutzdatenanalyse können dadurch entscheidende Informationen fehlen.

Bei ausreichendem Speicherplatz sollte eine bewusste Kürzung deshalb nur erfolgen, wenn der Anwendungsfall dies rechtfertigt.

Capture in Datei schreiben

Mit:

```
-w DATEI
```

werden die Rohpakete in eine Capture-Datei geschrieben, anstatt sie normal im Terminal zu dekodieren.

Beispiel:

```
sudo tcpdump -i INTERFACE -w diagnose.pcap
```

Diese Datei kann anschließend mit TCPDump oder Wireshark geöffnet werden.

`-w` speichert Pakete, nicht Terminaltext

Eine mit `-w` erzeugte Capture-Datei enthält die aufgezeichneten Paketdaten im Capture-Format.

Sie ist keine gewöhnliche Textdatei mit der sonst sichtbaren TCPDump-Ausgabe.

Das ist entscheidend:

Textausgabe → für unmittelbare menschliche Betrachtung

Capture-Datei → für erneute Paketverarbeitung und detaillierte Analyse

Capture-Datei lesen

Mit:

```
-r DATEI
```

wird eine vorhandene Capture-Datei gelesen.

Beispiel:

```
tcpdump -nn -r diagnose.pcap
```

Dafür ist keine Live-Netzwerkverbindung erforderlich.

Filter auf gespeicherte Captures anwenden

Auch beim Lesen einer vorhandenen Capture-Datei kann ein Filterausdruck verwendet werden.

Beispiel:

```
tcpdump -nn -r diagnose.pcap tcp port 443
```

Damit wird nicht die Datei verändert.

TCPDump zeigt lediglich die Pakete an, die dem Filter entsprechen.

Offline-Analyse

Die Kombination:

```
tcpdump -r
```

ist innerhalb eines Rescue-Systems besonders nützlich.

Ein Capture kann:

auf einem anderen System erzeugt,

auf ein Analysemedium kopiert

und anschließend vollständig offline untersucht

werden.

Dadurch kann die Analyse von der ursprünglichen Produktionsumgebung getrennt werden.

Wireshark-Übergabe

Ein typischer Arbeitsablauf lautet:

```
sudo tcpdump -i INTERFACE -w diagnose.pcap
```

Anschließend wird die Aufzeichnung beendet.

Die erzeugte Datei wird danach mit Wireshark geöffnet.

TCPDump übernimmt dabei die kompakte Erfassung.

Wireshark übernimmt anschließend die detaillierte grafische Protokollanalyse.

Terminal und GUI ergänzen sich

TCPDump ist besonders geeignet, wenn:

keine grafische Oberfläche benötigt wird,

ein schneller Capture gestartet werden soll,

remote gearbeitet wird,

nur bestimmte Pakete interessieren,

oder ein Capture für spätere Analyse erzeugt werden soll.

Wireshark ist dagegen besonders leistungsfähig bei:

Protokolldekodierung,

Streamanalyse,

Statistiken,

grafischer Navigation

und detaillierter Untersuchung einzelner Felder.

ASCII-Ausgabe

Mit:

`-A`

kann der Inhalt geeigneter Paketdaten zusätzlich als ASCII dargestellt werden.

Beispiel:

```
sudo tcpdump -A -nn tcp port 80
```

Dies kann bei unverschlüsselten textbasierten Protokollen hilfreich sein.

Es muss jedoch beachtet werden, dass Netzwerkdaten sensible Inhalte enthalten können.

Hex- und ASCII-Ausgabe

Mit:

`-X`

werden Paketdaten hexadezimal und zusätzlich als ASCII ausgegeben.

Dies ist nützlich, wenn sowohl die tatsächlichen Bytewerte als auch darstellbare Zeichen untersucht werden sollen.

Hex-Ausgabe

TCPDump besitzt weitere Ausgabeoptionen für hexadezimale beziehungsweise kombinierte Darstellungen.

Diese sind insbesondere bei Low-Level-Protokollanalysen nützlich.

Für große Captures ist eine vollständige Hexausgabe jedoch schnell sehr umfangreich.

Link-Layer-Header

Mit:

`-e`

wird der Link-Layer-Header in der Ausgabe dargestellt.

Bei Ethernet kann dies beispielsweise zusätzliche MAC-bezogene Informationen sichtbar machen.

Dies ist für ARP-, VLAN- und Layer-2-Diagnosen hilfreich.

Paketinhalt kann sensibel sein

Optionen wie:

`-A`

und

`-X`

können Nutzdaten direkt im Terminal sichtbar machen.

Bei unverschlüsselten Protokollen können darin beispielsweise:

Benutzernamen,

Anwendungsdaten,

URLs,

Header,

interne Informationen

oder andere vertrauliche Inhalte

enthalten sein.

Terminalausgaben und gespeicherte Logs sind daher entsprechend zu behandeln.

Richtung des Captures

TCPDump 4.99.5 unterstützt:

`-Q in`

`-Q out`

und:

`-Q inout`

zur Auswahl der Capture-Richtung, sofern die verwendete Capture-Plattform diese Funktion unterstützt.

Damit kann beispielsweise gezielt nur eingehender oder nur ausgehender Verkehr betrachtet werden.

Die Unterstützung ist plattformabhängig.

Eingehender Verkehr

Beispielprinzip:

```
sudo tcpdump -Q in -i INTERFACE
```

Damit wird – bei unterstützter Plattform – auf eingehende Pakete eingeschränkt.

Ausgehender Verkehr

Beispiel:

```
sudo tcpdump -Q out -i INTERFACE
```

Damit kann entsprechend ausgehender Verkehr untersucht werden.

Promiscuous Mode

TCPDump versucht auf geeigneten Interfaces standardmäßig, die Paketerfassung im Promiscuous Mode durchzuführen.

Mit:

```
-p
```

wird das Interface ausdrücklich **nicht** in den Promiscuous Mode versetzt.

Promiscuous Mode sieht nicht automatisch das gesamte LAN

Wie bei Wireshark gilt:

Promiscuous Mode bedeutet nicht, dass ein Rechner automatisch sämtlichen Unicast-Verkehr eines geswitchten Netzwerks empfängt.

Welche Pakete sichtbar sind, hängt davon ab, was die Netzwerkinfrastruktur tatsächlich zum Capture-Interface überträgt.

Für die Beobachtung fremden Unicast-Verkehrs kann ein entsprechend konfigurierter Mirror-/SPAN-Port oder ein anderer autorisierter Messpunkt erforderlich sein.

Capture-Puffer

TCPDump beziehungsweise libpcap verwendet einen Capture-Puffer.

Mit:

```
-B
```

kann dessen Größe auf unterstützten Plattformen eingestellt werden.

Bei sehr hohem Netzwerkaufkommen kann die Puffergröße Einfluss darauf haben, ob die Capture-Anwendung Pakete schnell genug verarbeitet.

Eine größere Puffergröße ist jedoch kein Ersatz für ausreichend leistungsfähige Hardware und eine geeignete Capture-Architektur.

Paketverluste

Beim Beenden eines Live-Captures gibt TCPDump Statistiken aus.

Dazu gehören unter anderem Angaben zu:

empfangenen Paketen

und

vom Kernel beziehungsweise Capture-Mechanismus verworfenen Paketen, soweit die Plattform diese Informationen bereitstellt.

Solche Drop-Zähler sind für die Bewertung eines Captures wichtig.

Ein unvollständiger Capture kann täuschen

Wenn während der Aufzeichnung Pakete verloren gehen, können beispielsweise:

TCP-Sequenzen unvollständig erscheinen,

Antworten fehlen,

Protokollabläufe falsch wirken

oder Retransmissions schwer interpretierbar werden.

Bei einer kritischen Analyse muss deshalb nicht nur der Netzwerkverkehr, sondern auch die Qualität der Aufzeichnung bewertet werden.

Rotation nach Dateigröße

Mit:

```
-C DATEIGRÖSSE
```

kann TCPDump beim Schreiben einer Capture-Datei automatisch auf eine neue Datei wechseln, wenn die aktuelle Datei die konfigurierte Größe erreicht.

Dies ist für Langzeitaufzeichnungen nützlich.

Rotation nach Zeit

Mit:

```
-G SEKUNDEN
```

kann eine neue Capture-Datei in zeitlichen Intervallen erzeugt werden.

Damit können beispielsweise regelmäßige Capture-Segmente erstellt werden.

Anzahl der Dateien begrenzen

Mit:

```
-W ANZAHL
```

kann in Verbindung mit Rotationsoptionen die Zahl erzeugter Dateien begrenzt werden.

Abhängig von der verwendeten Kombination können Dateien zyklisch wiederverwendet beziehungsweise die Erfassung nach Erreichen der Grenze beendet werden.

Vor einem Langzeit-Capture sollte die genaue gewünschte Rotationslogik daher bewusst festgelegt werden.

Langzeit-Captures

Für längere Aufzeichnungen sollte vorab geklärt werden:

welcher Verkehr benötigt wird,

wie groß die Datenrate ist,

wie viel Speicherplatz verfügbar ist,

wie lange aufgezeichnet werden soll,

ob Dateien rotiert werden,

und ob ältere Daten überschrieben werden dürfen.

Ein unkontrollierter Capture kann ansonsten einen Datenträger vollständig füllen.

Sofortige Ausgabe

TCPDump puffert seine Ausgabe.

Für bestimmte Diagnose- und Weiterverarbeitungsszenarien stehen Optionen zur Steuerung des Output-Bufferings zur Verfügung.

Dies kann insbesondere bei Pipelines oder einer Live-Auswertung relevant sein.

Capture über Pipe

TCPDump kann Capture-Daten über die Standardausgabe weitergeben, wenn `-w -` verwendet wird.

Dies ermöglicht Unix-Pipelines und spezialisierte Weiterverarbeitung.

Solche Konstruktionen sollten nur eingesetzt werden, wenn die Datenwege und Zielprogramme genau bekannt sind.

Keine Paketinjektion

TCPDump ist primär ein Capture- und Analysewerkzeug.

Seine normale Aufgabe ist nicht, beliebige Netzwerkpakete zu erzeugen oder zu injizieren.

Damit unterscheidet es sich beispielsweise von Nmap, das aktiv Netzwerkprobes erzeugt.

TCPDump ist überwiegend passiv

Ein normaler TCPDump-Capture erzeugt nicht selbst die zu untersuchende Anwendungskommunikation.

Er beobachtet den am Capture-Punkt sichtbaren Verkehr.

Es können jedoch indirekte Effekte entstehen, beispielsweise wenn Namensauflösung für die Ausgabe verwendet wird.

Auch deshalb ist `-n` beziehungsweise `-nn` für kontrollierte Diagnosen häufig sinnvoll.

TCPDump und Nmap

Nmap und TCPDump ergänzen sich unmittelbar.

Ein möglicher Ablauf lautet:

TCPDump-Capture starten.

Nmap-Scan durchführen.

TCPDump-Capture stoppen.

Capture anschließend analysieren.

Damit lässt sich auf Paketebene nachvollziehen:

welche Probes Nmap gesendet hat,

welche Antworten zurückkamen

und warum ein Port möglicherweise als offen, geschlossen oder gefiltert bewertet wurde.

TCPDump und Wireshark

Beide Werkzeuge können Capture-Dateien verarbeiten.

Die Aufgabenteilung innerhalb von JUMBO BLOCK Rescue kann vereinfacht beschrieben werden als:

TCPDump → schnell erfassen und filtern

Wireshark → detailliert grafisch analysieren

TCPDump ist dabei nicht lediglich eine „kleinere Version“ von Wireshark.

Seine Stärke liegt insbesondere in der direkten Einbindung in Terminal-, Remote- und automatisierte Arbeitsabläufe.

TCPDump und ARP Scan

ARP Scan erzeugt aktiv ARP-Anfragen, um Geräte im lokalen Netzwerk zu entdecken.

TCPDump kann diesen ARP-Verkehr beobachten.

Beispielsweise:

```
sudo tcpdump -nn arp
```

Damit lässt sich nachvollziehen, welche ARP-Anfragen gesendet und welche Antworten empfangen werden.

TCPDump und MTR

MTR erzeugt Diagnoseverkehr zur Untersuchung des Netzwerkpfads.

TCPDump kann diesen Verkehr parallel aufzeichnen.

Dies kann bei ungewöhnlichen Routing- oder Firewall-Problemen zusätzliche Informationen liefern.

TCPDump und Dig

Dig erzeugt gezielte DNS-Abfragen.

TCPDump kann gleichzeitig den tatsächlichen DNS-Verkehr beobachten.

Ein typischer Filter ist:

```
sudo tcpdump -nn port 53
```

Damit lässt sich prüfen:

ob die Anfrage tatsächlich gesendet wird,

an welche Adresse sie geht,

ob eine Antwort zurückkommt

und welche Transportkommunikation stattfindet.

TCPDump und GUFW

GUFW/UFW kann Pakete entsprechend der Firewall-Regeln filtern.

TCPDump beobachtet Netzwerkverkehr an einem bestimmten Capture-Punkt im Netzwerkstack.

Bei der Interpretation eines Captures muss deshalb berücksichtigt werden, an welcher Stelle die Aufzeichnung erfolgt und welche Firewall-Regeln aktiv sind.

Die Abwesenheit eines erwarteten Pakets im Capture beweist nicht ohne weitere Analyse, an welcher Stelle des Kommunikationswegs es verloren ging.

Typischer Rescue-Ablauf

Ein kontrollierter TCPDump-Einsatz kann folgendermaßen aussehen:

1. Interfaces bestimmen

```
tcpdump -D
```

2. Relevantes Interface auswählen

3. Namensauflösung deaktivieren

Für klare technische Ausgabe:

```
-nn
```

4. Problem möglichst gezielt filtern

Beispielsweise nach Host, Port oder Protokoll.

5. Capture zeitlich oder nach Paketanzahl begrenzen

Beispielsweise mit:

`-c`

oder durch bewusstes manuelles Stoppen.

6. Bei komplexen Problemen Capture-Datei erzeugen

`-w diagnose.pcap`

7. Problem reproduzieren

8. Capture stoppen

9. Datei zunächst mit TCPDump oder anschließend mit Wireshark analysieren

10. Original-Capture bei dokumentations- oder beweisrelevanten Untersuchungen unverändert erhalten

Beispiel: DNS funktioniert nicht

Capture:

```
sudo tcpdump -nn -i INTERFACE port 53
```

Anschließend wird die fehlerhafte DNS-Abfrage ausgelöst.

Nun kann geprüft werden:

Geht eine Anfrage hinaus?

An welche IP-Adresse?

Über UDP oder TCP?

Kommt eine Antwort zurück?

Falls keine Antwort sichtbar ist, muss anschließend weiter untersucht werden, ob das Problem beispielsweise beim DNS-Server, Routing oder einer Filterung liegt.

Beispiel: TCP-Verbindung

Für einen konkreten Dienst:

```
sudo tcpdump -nn -i INTERFACE tcp port PORT
```

Anschließend wird die Verbindung ausgelöst.

Nun kann untersucht werden, ob:

SYN gesendet,

SYN/ACK empfangen,

RST empfangen

oder überhaupt keine Antwort sichtbar

wird.

Damit lässt sich ein Verbindungsproblem auf grundlegender TCP-Ebene untersuchen.

Beispiel: ARP-Problem

```
sudo tcpdump -nn -i INTERFACE arp
```

Anschließend wird beobachtet, ob eine ARP-Anfrage für die benötigte IPv4-Adresse gesendet und beantwortet wird.

Fehlt bereits die ARP-Auflösung im lokalen Ethernet, kann eine IP-basierte Verbindung zum betreffenden lokalen Ziel nicht normal aufgebaut werden.

Beispiel: Capture für Wireshark

```
sudo tcpdump -nn -i INTERFACE -w diagnose.pcap
```

Problem reproduzieren.

Capture beenden.

Danach:

```
wireshark diagnose.pcap
```

Damit wird die Aufzeichnung in Wireshark für eine detaillierte grafische Analyse geöffnet.

Beispiel: Nur begrenzte Paketmenge

```
sudo tcpdump -nn -i INTERFACE -c 500 -w diagnose.pcap
```

Der Capture wird nach 500 erfassten Paketen beendet.

Dies kann bei reproduzierbaren kurzen Fehlerbildern verhindern, dass unnötig große Dateien entstehen.

Capture-Dateien als sensible Daten

Wie Wireshark-PCAPs können TCPDump-Captures sensible Informationen enthalten.

Dazu gehören abhängig vom Verkehr:

IP-Adressen,

MAC-Adressen,

Hostnamen,

DNS-Abfragen,

Kommunikationsbeziehungen,

unverschlüsselte Nutzdaten,

Anmeldedaten unsicherer Protokolle

und andere interne Informationen.

Capture-Dateien sind daher entsprechend ihrer Inhalte zu schützen.

Datenschutz und Berechtigung

Netzwerkverkehr darf nur dort aufgezeichnet werden, wo die notwendige technische und organisatorische Berechtigung besteht.

Ein Capture kann Kommunikation anderer Benutzer oder Systeme enthalten.

Dies gilt besonders für:

Unternehmensnetze,

KRITIS,

Produktionsnetze,

administrative Netze

und gemeinsam genutzte Infrastrukturen.

Verschlüsselter Verkehr

TCPDump kann verschlüsselte Netzwerkpakete aufzeichnen.

Die Verschlüsselung des Nutzdateninhalts wird dadurch jedoch nicht automatisch aufgehoben.

Bei TLS beispielsweise bleiben die verschlüsselten Nutzdaten ohne die erforderlichen kryptografischen Informationen verschlüsselt.

Metadaten wie:

IP-Adressen,

Ports,

Paketgrößen

und Zeitpunkte

bleiben dennoch sichtbar.

Grenzen

TCPDump sieht nur den Verkehr, der am gewählten Capture-Punkt verfügbar ist.

Promiscuous Mode liefert nicht automatisch den gesamten Verkehr eines geswitchten Netzwerks.

Ein Capture Filter verwirft nicht passende Pakete bereits während der Erfassung.

Eine zu kleine Snaplen kann wichtige Paketdaten abschneiden.

Paketverluste während des Captures können die Analyse verfälschen.

Verschlüsselte Nutzdaten bleiben ohne geeignete Schlüssel beziehungsweise Session-Informationen verschlüsselt.

TCPDump ersetzt keine vollständige grafische Protokollanalyse.

Es ist kein Portscanner, kein Malware-Scanner und kein Intrusion-Prevention-System.

Sicherheitshinweise

Live-Captures nur in autorisierten Netzwerken durchführen.

Für technische Diagnose möglichst `-n` beziehungsweise `-nn` verwenden, wenn keine Namensauflösung benötigt wird.

Capture Filter nicht unnötig eng setzen, wenn die Fehlerursache noch unbekannt ist.

Speicherplatz vor Langzeitaufzeichnungen prüfen.

Bei Rotationen genau festlegen, ob alte Capture-Dateien überschrieben werden dürfen.

Capture-Drop-Zähler bei kritischen Analysen berücksichtigen.

PCAP-Dateien als potenziell sensible Daten behandeln.

Unverschlüsselte Paketdaten nicht unkontrolliert protokollieren oder weitergeben.

Bei beweisrelevanten Untersuchungen Original-Capture unverändert erhalten und weitere Verarbeitung an Kopien durchführen.

Technische Einordnung in JUMBO BLOCK Rescue

TCPDump bildet innerhalb von JUMBO BLOCK Rescue die **terminalbasierte Ebene der Paketaufzeichnung und Netzwerkverkehrsanalyse**.

Die ersten drei Werkzeuge des Netzwerkbereichs ergeben damit eine klare technische Staffelung:

A.17 Nmap

aktive Netzwerkerkennung, Portanalyse, Dienst- und Systemklassifikation.

A.18 Wireshark

umfangreiche grafische Paket- und Protokollanalyse.

A.19 TCPDump

schnelle terminalbasierte Paketaufzeichnung, Filterung und Erstellung von Capture-Dateien.

Ein besonders wichtiger Rescue-Workflow lautet:

TCPDump zeichnet auf → PCAP wird gesichert → Wireshark analysiert detailliert.

Ebenso kann:

Nmap erzeugt Diagnoseverkehr → TCPDump zeichnet ihn auf → Wireshark erklärt ihn auf Protokollebene

als kombinierter Analyseablauf verwendet werden.

TCPDump ist damit das kompakte Capture-Werkzeug von JUMBO BLOCK Rescue für Situationen, in denen eine schnelle, kontrollierte und unmittelbar im Terminal verfügbare Sicht auf den Netzwerkverkehr erforderlich ist.

A.20 ARP Scan

Bereich: Netzwerk
Programm: arp-scan
Baseline 20261001: arp-scan 1.10.0
Installiertes Paket: `arp-scan 1.10.0-2+b1`
Bedienung: Kommandozeile
Aufgabe: Erkennung und grundlegende Identifikation aktiver IPv4-Geräte im lokalen Layer-2-Netz mittels ARP

Zweck

ARP Scan ist ein spezialisiertes Werkzeug zur Erkennung von IPv4-Systemen im lokalen Netzwerk.

Das Programm sendet gezielt ARP-Requests an die angegebenen IPv4-Ziele und zeigt die empfangenen ARP-Antworten an.

Dadurch kann ARP Scan Geräte im unmittelbar erreichbaren lokalen Layer-2-Netz erkennen.

Typische Ergebnisse enthalten:

IPv4-Adresse,

MAC-Adresse

und – sofern eine Zuordnung möglich ist – Hersteller- beziehungsweise Vendorinformationen zur MAC-Adresse.

ARP Scan ist damit wesentlich spezialisierter als Nmap.

Seine zentrale Frage lautet:

Welche IPv4-Geräte antworten im lokalen Netzwerk auf ARP?

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich ARP Scan insbesondere für:

schnelle Bestandsaufnahme eines lokalen IPv4-Netzes,

Erkennung unbekannter Geräte,

Ermittlung von IPv4-/MAC-Zuordnungen,

Suche nach Geräten, deren IP-Adresse unbekannt ist,

Kontrolle, ob ein erwartetes Gerät lokal erreichbar ist,

Erkennung von Systemen trotz blockierter ICMP-Echo-Anfragen,

Ermittlung eines möglichen Geräteherstellers anhand der MAC-Adresse,

und Vorbereitung einer anschließenden detaillierteren Untersuchung mit Nmap, Wireshark oder TCPDump.

ARP

ARP steht für:

Address Resolution Protocol

In IPv4-Ethernet-Netzen wird ARP verwendet, um zu einer lokalen IPv4-Adresse die zugehörige Link-Layer-Adresse – bei Ethernet typischerweise eine MAC-Adresse – zu ermitteln.

Ein System kann beispielsweise sinngemäß fragen:

Wer besitzt diese IPv4-Adresse?

Das angesprochene System kann darauf mit seiner MAC-Adresse antworten.

ARP Scan automatisiert solche ARP-Anfragen für eine definierte Menge von Zieladressen.

Grundprinzip

ARP Scan sendet für jedes Ziel einen ARP Request.

Antwortet das Ziel, wird die Antwort ausgewertet und angezeigt.

Dadurch entsteht eine Liste der antwortenden Geräte im lokalen Netzwerk.

ARP arbeitet nur lokal

Eine der wichtigsten Grenzen von ARP Scan lautet:

ARP wird nicht geroutet.

ARP Scan kann deshalb nicht einfach Geräte hinter einem Router auf dieselbe Weise untersuchen.

Das Werkzeug ist für das unmittelbar über das betreffende Layer-2-Netz erreichbare lokale Netzwerk vorgesehen.

Konsequenz für den Rescue-Einsatz

Befindet sich beispielsweise ein Router zwischen JUMBO BLOCK Rescue und einem Zielnetz, werden die ARP-Anfragen nicht durch den Router in dieses entfernte Netz weitergeleitet.

Für entfernte IP-Netze sind andere Werkzeuge und Verfahren erforderlich.

Nmap kann beispielsweise IP-basierte Untersuchungen über Routing-Grenzen hinweg durchführen, sofern das Zielnetz erreichbar und die Untersuchung autorisiert ist.

IPv4

ARP Scan dient der Erkennung von **IPv4-Hosts**.

ARP selbst verwendet IPv4-Adressen zur Zuordnung.

IPv6 verwendet für die entsprechende Nachbarschaftserkennung nicht ARP, sondern Mechanismen des Neighbor Discovery Protocol.

ARP Scan ist deshalb kein allgemeiner IPv6-Hostscanner.

Grundsyntax

Die grundlegende Syntax lautet:

```
arp-scan [Optionen] [Ziele]
```

Ziele können unter anderem als:

IPv4-Adresse,

Hostname,

CIDR-Netz,

Adressbereich

oder Netzwerk mit Netzmaske

angegeben werden.

Einzelnes Ziel

Beispiel:

```
sudo arp-scan 192.0.2.10
```

Die Adresse dient ausschließlich als Dokumentationsbeispiel.

ARP Scan sendet dabei eine ARP-Anfrage für das angegebene Ziel.

Mehrere Ziele

Mehrere Zieladressen können angegeben werden.

Damit kann eine definierte Gruppe lokaler Systeme untersucht werden.

CIDR-Netz

Ein Netzwerk kann in CIDR-Notation angegeben werden.

Beispiel:

```
sudo arp-scan 192.0.2.0/24
```

ARP Scan erzeugt daraus die entsprechenden Zieladressen.

Bei der CIDR-Zieldefinition werden auch Netzwerk- und Broadcastadresse in die Zielliste einbezogen.

Adressbereich

ARP Scan unterstützt auch explizite IPv4-Adressbereiche.

Beispielprinzip:

```
192.0.2.1-192.0.2.20
```

Damit kann ein kleiner Teil eines lokalen Netzes gezielt untersucht werden.

Netzwerk und Netzmaske

Alternativ unterstützt ARP Scan die Schreibweise:

```
Netzwerk:Netzmaske
```

Beispielprinzip:

```
192.0.2.0:255.255.255.0
```

Damit kann ebenfalls eine Zielmenge definiert werden.

Lokales Netz automatisch bestimmen

Eine der wichtigsten Optionen lautet:

```
--localnet
```

Kurzform:

```
-l
```

Beispiel:

```
sudo arp-scan --localnet
```

ARP Scan erzeugt dabei die Zielliste anhand der IPv4-Adresse und Netzmaske des verwendeten Interfaces.

Bedeutung von `--localnet`

`--localnet` bedeutet nicht:

„Scanne sämtliche Netze, die dieser Rechner erreichen kann.“

Es bedeutet:

Erzeuge die Zieladressen aus der IPv4-Konfiguration des verwendeten lokalen Interfaces.

Das ist für den Rescue-Einsatz eine wichtige Abgrenzung.

Netzwerk- und Broadcastadresse

Bei `--localnet` werden die Adressen aus Interface-Adresse und Netzmaske generiert.

Die daraus erzeugte Liste enthält auch Netzwerk- und Broadcastadresse.

ARP Scan arbeitet damit die erzeugte Zielmenge systematisch ab.

Interface auswählen

Mit:

`--interface=INTERFACE`

beziehungsweise:

`-I INTERFACE`

wird das zu verwendende Netzwerkinterface ausdrücklich festgelegt.

Beispiel:

```
sudo arp-scan --interface=INTERFACE --localnet
```

Dies ist besonders wichtig, wenn ein Rescue-System mehrere aktive Netzwerkinterfaces besitzt.

Automatische Interface-Auswahl

Wird kein Interface angegeben, sucht ARP Scan nach dem niedrigsten nummerierten konfigurierten und aktiven Interface, wobei Loopback ausgeschlossen wird.

Für reproduzierbare technische Untersuchungen ist es dennoch sinnvoll, das gewünschte Interface ausdrücklich anzugeben, wenn mehrere Interfaces vorhanden sind.

Interface ohne IPv4-Adresse

ARP Scan kann grundsätzlich auch auf einem Interface verwendet werden, das keine eigene IPv4-Adresse besitzt.

ARP verwendet zwar IPv4-Adressen innerhalb seiner Nachrichten, ist selbst jedoch kein IP-Protokoll, das durch Router transportiert wird.

Für `--localnet` wird dagegen eine Interface-Konfiguration benötigt, aus der die lokale Zielmenge erzeugt werden kann.

Berechtigungen

ARP Scan verwendet Raw-Socket-Funktionen.

Dafür sind entsprechende Berechtigungen erforderlich.

Unter Linux unterstützt Version 1.10.0 POSIX.1e Capabilities und benötigt für die entsprechenden Funktionen `CAP_NET_RAW`.

Alternativ kann eine Ausführung mit ausreichenden Systemrechten erfolgen.

Welche konkrete Rechtekonfiguration JUMBO BLOCK Rescue verwendet, wird aus der bloßen Paketbaseline nicht abgeleitet.

Die Beispiele verwenden daher bei privilegiertem Zugriff:

`sudo`

Standardausgabe

Für antwortende Hosts zeigt ARP Scan typischerweise:

IPv4-Adresse,

MAC-Adresse

und Vendorinformation

an.

Beispielstruktur:

```
IPv4-Adresse MAC-Adresse Hersteller
```

Die konkrete Ausgabe hängt von den vorhandenen Vendor-Daten und den gewählten Optionen ab.

MAC-Adresse

Die MAC-Adresse identifiziert die Netzwerkschnittstelle auf der lokalen Link-Layer-Ebene.

Sie ist für die lokale Geräteanalyse besonders hilfreich, weil eine IPv4-Adresse wechseln kann, während die MAC-Adresse einer physischen Schnittstelle häufig stabiler bleibt.

Sie darf dennoch nicht als unveränderliche Geräteidentität betrachtet werden.

MAC-Adressen können geändert werden

MAC-Adressen können:

softwareseitig verändert,

randomisiert,

virtuell erzeugt,

durch Virtualisierung bereitgestellt

oder bewusst nachgebildet

werden.

Eine MAC-Adresse allein beweist daher nicht die physische Identität eines Geräts.

Vendor-Erkennung

ARP Scan kann anhand der MAC-Adresse versuchen, einen Hersteller beziehungsweise eine registrierte Organisation zuzuordnen.

Version 1.10.0 verwendet hierfür insbesondere:

`ieee-oui.txt`

und:

`mac-vendor.txt`.

IEEE-Zuordnungen

Die Datei:

`ieee-oui.txt`

enthält in Version 1.10.0 Zuordnungen aus verschiedenen IEEE-MAC-Adressregistern.

Dazu gehören:

MA-L beziehungsweise OUI,

MA-M,

MA-S beziehungsweise OUI36

und IAB.

Zusätzliche Zuordnungen

`mac-vendor.txt`

enthält zusätzliche MAC-/Vendor-Zuordnungen und kann auch für lokale beziehungsweise benutzerdefinierte Einträge verwendet werden.

Herstellerangabe richtig interpretieren

Die Vendorinformation ist ein **Hinweis auf die registrierte MAC-Adresszuordnung**.

Sie ist nicht zwingend der Hersteller des vollständigen Geräts.

Beispielsweise kann eine Netzwerkschnittstelle beziehungsweise ein eingebauter Netzwerkchip von einem anderen Hersteller stammen als das Gesamtsystem.

Virtualisierung

Bei virtuellen Maschinen kann die MAC-Adresse einem Virtualisierungsanbieter beziehungsweise einem dafür verwendeten Adressbereich zugeordnet sein.

Eine Vendoranzeige kann deshalb beispielsweise Hinweise auf eine virtuelle Netzwerkschnittstelle liefern, ohne den tatsächlichen Hersteller des physischen Hosts zu bestimmen.

Keine Vendorinformation

Kann eine MAC-Adresse keiner bekannten Zuordnung zugeordnet werden, bedeutet dies nicht, dass das Gerät unbekannt oder fehlerhaft ist.

Mögliche Ursachen sind unter anderem:

lokal administrierte MAC-Adresse,

randomisierte MAC-Adresse,

fehlender Eintrag in den verwendeten Daten,

oder eine nicht passende Vendor-Zuordnung.

OUI-Datei aktualisieren

Zum arp-scan-Projekt gehört das Hilfsprogramm:

`get-oui`

mit dem die IEEE-basierten Vendorinformationen aktualisiert werden können.

Ein solcher Updatevorgang benötigt eine entsprechende Datenquelle beziehungsweise Netzwerkverbindung.

Für die eingefrorene Rescue-Baseline wird daraus nicht abgeleitet, dass die Vendor-Daten automatisch aktualisiert werden.

Minimalistische Ausgabe

Mit:

`--quiet`

beziehungsweise:

`-q`

wird eine reduzierte Ausgabe verwendet.

Dabei werden für antwortende Hosts nur:

IP-Adresse

und

MAC-Adresse

angezeigt.

Die Vendor-Daten werden in diesem Modus nicht für die normale Ausgabe benötigt.

Nur Hostzeilen

Mit:

`--plain`

beziehungsweise:

`-x`

werden Header- und Footerinformationen unterdrückt.

Es werden nur die Informationen der antwortenden Hosts ausgegeben.

Dies ist insbesondere für eine spätere Verarbeitung durch Skripte nützlich.

Benutzerdefiniertes Ausgabeformat

Version 1.10.0 führte die Option:

`--format`

ein.

Damit kann das Ausgabeformat flexibel definiert werden.

Dies erleichtert eine strukturierte Weiterverarbeitung der Scanergebnisse.

Hostnamen auflösen

Mit:

`--resolve`

beziehungsweise:

`-d`

können die IP-Adressen antwortender Systeme in Hostnamen aufgelöst werden.

Eine solche Namensauflösung erzeugt beziehungsweise benötigt jedoch zusätzliche Namensauflösungsaktivität und kann die Ausgabe beeinflussen.

Für eine reine Layer-2-Bestandsaufnahme sind numerische IPv4- und MAC-Adressen häufig eindeutiger.

Nur numerische Zieladressen

Mit:

`--numeric`

beziehungsweise:

`-N`

wird festgelegt, dass Ziele IPv4-Adressen und keine Hostnamen sein müssen.

Bei großen Ziellisten kann dadurch der Aufwand für die Namensauflösung beim Start reduziert werden.

Ziele aus Datei

Mit:

`--file=DATEI`

beziehungsweise:

`-f DATEI`

können Zieladressen beziehungsweise Zielmuster aus einer Datei gelesen werden.

Pro Zeile wird ein entsprechendes Ziel angegeben.

Mit:

`-`

kann die Standardeingabe verwendet werden.

`--file` und `--localnet`

Bei Verwendung von:

`--localnet`

können nicht gleichzeitig Ziele über `--file` oder zusätzlich über die Kommandozeile vorgegeben werden.

`--localnet` erzeugt selbst die zu scannende Zielmenge aus der Interface-Konfiguration.

Wiederholungsversuche

ARP Scan sendet standardmäßig nicht unbegrenzt Anfragen.

Mit:

`--retry=ANZAHL`

beziehungsweise:

`-r ANZAHL`

wird die Gesamtzahl der Versuche pro Host festgelegt.

Der dokumentierte Standardwert beträgt:

2 Versuche pro Host.

Warum Wiederholungen?

ARP arbeitet lokal und ist normalerweise sehr zuverlässig.

Trotzdem können Pakete beispielsweise aufgrund von:

Netzlast,

Paketverlust,

überlasteten Geräten

oder anderen temporären Bedingungen

verloren gehen.

Ein Wiederholungsversuch reduziert die Wahrscheinlichkeit, einen aktiven Host nur wegen eines einzelnen verlorenen Pakets zu übersehen.

Weniger Wiederholungen

Weniger Wiederholungen verkürzen den Scan.

Gleichzeitig steigt das Risiko, ein tatsächlich vorhandenes Gerät bei Paketverlust nicht zu erkennen.

Timeout

Mit:

`--timeout=MILLISEKUNDEN`

beziehungsweise:

`-t`

wird der anfängliche Timeout pro Host festgelegt.

Der dokumentierte Standardwert beträgt:

500 ms.

Backoff

Mit:

`--backoff=FAKTOR`

beziehungsweise:

`-b`

wird der Backoff-Faktor für nachfolgende Durchläufe festgelegt.

Der dokumentierte Standardwert beträgt:

1,5.

Damit können spätere Versuche längere Antwortzeiten berücksichtigen.

Bandbreite

Mit:

`--bandwidth=WERT`

beziehungsweise:

`-B`

wird die gewünschte ausgehende Scanbandbreite festgelegt.

Der dokumentierte Standardwert beträgt:

256.000 Bit/s.

Warum Bandbreite begrenzen?

ARP Scan erzeugt aktiv ARP-Anfragen.

Bei großen Zielbereichen kann dadurch eine erhebliche Zahl von Broadcast-Anfragen entstehen.

Eine kontrollierte Senderate verhindert unnötig hohe Last.

Höhere Bandbreite

Eine höhere Bandbreite kann einen Scan beschleunigen.

Zu hohe Werte können jedoch eine hohe Broadcast-Paketrate verursachen.

In empfindlichen Produktions-, Industrie- oder KRITIS-Netzen sollte die Scanrate deshalb nicht ohne technischen Grund erhöht werden.

Paketintervall

Alternativ kann mit:

`--interval`

beziehungsweise:

`-i`

ein minimales Intervall zwischen ausgehenden Paketen festgelegt werden.

Damit lässt sich die Senderate unmittelbar beeinflussen.

`--interval` und `--bandwidth` werden nicht gleichzeitig verwendet.

Duplikate

Ein Host kann mehr als einmal antworten.

ARP Scan kann solche doppelten Antworten erkennen und entsprechend darstellen.

Mit:

`--ignoredups`

beziehungsweise:

`-g`

kann die Anzeige doppelter Pakete unterdrückt werden.

Doppelte Antworten können relevant sein

Duplikate sind nicht grundsätzlich bedeutungslos.

Wenn beispielsweise mehrere Antworten auf eine Zieladresse mit unterschiedlichen MAC-Adressen auftreten, kann dies auf eine besondere Netzwerkkonstellation hinweisen.

Eine solche Beobachtung sollte untersucht und nicht vorschnell als Angriff interpretiert werden.

Mögliche Ursachen können unter anderem legitime Hochverfügbarkeits-, Virtualisierungs- oder Netzwerkkonfigurationen sein.

ARP und Firewalls

ARP arbeitet unterhalb der normalen IP-Portebene.

Ein System, das ICMP-Echo oder TCP-Verbindungsversuche filtert, kann trotzdem auf ARP reagieren.

Dadurch ist ARP Scan für die lokale Host-Erkennung besonders nützlich.

Ein nicht pingbares Gerät kann sichtbar sein

Ein Gerät kann beispielsweise:

ICMP Echo Request blockieren

und

trotzdem auf ARP Requests antworten.

Deshalb ist:

„Ping antwortet nicht“

nicht gleichbedeutend mit:

„Gerät ist im lokalen Netz nicht vorhanden.“

Ein ARP-Fund beweist keinen Dienst

Eine ARP-Antwort zeigt, dass auf der lokalen Netzebene ein System auf die betreffende IPv4-Adresse reagiert.

Sie sagt jedoch nichts darüber aus:

welche TCP-Ports offen sind,

welche UDP-Dienste vorhanden sind,

welches Betriebssystem läuft,

oder ob ein bestimmter Anwendungsdienst funktioniert.

Dafür sind weitere Werkzeuge erforderlich.

ARP Scan und Nmap

Nmap besitzt ebenfalls ARP-basierte Host-Discovery-Funktionen für lokale Ethernet-Ziele.

ARP Scan ist jedoch speziell auf diese Aufgabe ausgerichtet.

Eine sinnvolle Arbeitsteilung lautet:

ARP Scan → lokale IPv4-Geräte schnell erkennen

Nmap → gefundene Systeme anschließend gezielt auf Ports, Dienste und weitere Netzwerkmerkmale untersuchen

ARP Scan und Wireshark

Wireshark kann die von ARP Scan erzeugten ARP-Pakete beobachten.

Ein Display Filter:

```
arp
```

zeigt den entsprechenden ARP-Verkehr.

Damit kann unmittelbar untersucht werden:

welche ARP Requests gesendet wurden,

welche Geräte geantwortet haben

und welche Adressinformationen in den Paketen enthalten sind.

ARP Scan und TCPDump

TCPDump kann denselben Vorgang terminalbasiert beobachten.

Beispiel:

```
sudo tcpdump -nn arp
```

Parallel kann ARP Scan ausgeführt werden.

Damit lassen sich gesendete Requests und empfangene Replies auf Paketebene nachvollziehen.

ARP Scan und MTR

MTR untersucht IP-basierte Netzwerkpfade über mehrere Router beziehungsweise Hops.

ARP Scan arbeitet dagegen ausschließlich im lokalen Layer-2-Netz.

Damit liegen die beiden Werkzeuge auf unterschiedlichen Analyseebenen:

ARP Scan → lokaler Nachbar

MTR → gerouteter Netzwerkpfad

Typischer Rescue-Ablauf

Ein kontrollierter Einsatz kann folgendermaßen erfolgen:

1. Physische beziehungsweise logische Netzwerkverbindung herstellen

2. Gewünschtes Interface bestimmen

Bei mehreren Interfaces das tatsächlich relevante Interface auswählen.

3. Lokales Netz scannen

Beispiel:

```
sudo arp-scan --interface=INTERFACE --localnet
```

4. Ergebnisse dokumentieren

Insbesondere:

IPv4-Adresse,

MAC-Adresse,

Vendorinformation.

5. Erwartete und unerwartete Systeme vergleichen

6. Interessante Systeme gezielt weiter untersuchen

Beispielsweise mit Nmap.

7. Bei Bedarf Paketverkehr beobachten

Mit Wireshark oder TCPDump.

Beispiel: IP-Adresse eines Geräts unbekannt

Ein Gerät befindet sich physisch im selben lokalen Ethernet-Netz, seine IPv4-Adresse ist jedoch unbekannt.

ARP Scan kann das lokale Netz untersuchen:

```
sudo arp-scan --localnet
```

Die antwortenden Geräte werden mit IPv4- und MAC-Adresse ausgegeben.

Anhand:

bekannter MAC-Adresse,

Vendorinformation

oder Vergleich mit anderen Geräten

kann das gesuchte System möglicherweise eingegrenzt werden.

Beispiel: Gerät antwortet nicht auf Ping

Ein erwartetes Gerät reagiert nicht auf ICMP Echo Requests.

Das beweist noch nicht, dass es offline ist.

Ein lokaler ARP-Scan kann prüfen, ob das System trotzdem auf ARP reagiert.

Antwortet es auf ARP, besteht zumindest lokale Layer-2-Kommunikation.

Die Ursache des fehlenden Ping muss dann auf einer anderen Ebene gesucht werden.

Beispiel: Unbekanntes Gerät

Ein ARP Scan zeigt eine unerwartete IPv4-/MAC-Kombination.

Das weitere Vorgehen sollte kontrolliert erfolgen:

MAC-/Vendorinformation dokumentieren.

Netzwerkdokumentation vergleichen.

Bei vorhandener Berechtigung das System anschließend gezielt mit Nmap untersuchen.

Falls erforderlich ARP- und IP-Kommunikation mit Wireshark oder TCPDump beobachten.

Ein unbekannter ARP-Eintrag allein beweist keinen Sicherheitsvorfall.

Beispiel: Falsches Interface

Besitzt JUMBO BLOCK Rescue mehrere Interfaces, kann ein Scan über das falsche Interface ein völlig anderes oder leeres Ergebnis liefern.

Deshalb sollte bei Unklarheit das Interface ausdrücklich angegeben werden:

```
sudo arp-scan -I INTERFACE --localnet
```

Damit ist eindeutig festgelegt, welches lokale Netz untersucht wird.

Grenzen

ARP Scan arbeitet mit ARP und damit mit IPv4-bezogener lokaler Nachbarschaftserkennung.

ARP wird nicht geroutet.

Entfernte Netze hinter Routern können damit nicht direkt auf dieselbe Weise untersucht werden.

ARP Scan führt keinen TCP-/UDP-Portscan durch.

Es erkennt keine Netzwerkdienste.

Es führt keine Betriebssystemerkennung wie Nmap durch.

Eine Vendorinformation ist keine sichere Geräteidentifikation.

MAC-Adressen können verändert, randomisiert oder virtualisiert sein.

Ein nicht antwortendes Gerät ist nicht mit absoluter Sicherheit ausgeschaltet; Paketverlust, Netzsegmentierung oder andere technische Ursachen können eine Antwort verhindern.

Sicherheitshinweise

Nur autorisierte lokale Netze untersuchen.

Vor dem Scan das richtige Interface und den vorgesehenen Netzbereich kontrollieren.

In Produktions- und KRITIS-Netzen Scanrate nicht unnötig erhöhen.

Vendorinformationen nur als Hinweis verwenden.

MAC-Adressen nicht als kryptografisch verlässliche Geräteidentität behandeln.

Unbekannte Geräte nicht allein aufgrund eines ARP-Ergebnisses als kompromittiert klassifizieren.

Bei ungewöhnlichen oder widersprüchlichen Ergebnissen mit Wireshark oder TCPDump auf Paketebene nachprüfen.

Technische Einordnung in JUMBO BLOCK Rescue

ARP Scan übernimmt innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue eine klar abgegrenzte Aufgabe:

schnelle aktive Erkennung lokaler IPv4-Geräte auf Layer 2.

Die bisherigen Netzwerkwerkzeuge bilden damit vier unterschiedliche Analyseperspektiven:

A.17 Nmap

aktive Host-, Port-, Dienst- und Systemanalyse.

A.18 Wireshark

grafische Paket- und Protokollanalyse.

A.19 TCPDump

terminalbasierte Paketaufzeichnung und Filterung.

A.20 ARP Scan

spezialisierte lokale IPv4-Geräteerkennung über ARP.

Ein besonders sinnvoller Rescue-Workflow lautet:

ARP Scan → lokales Gerät finden

Nmap → Ports und Dienste des gefundenen Geräts untersuchen

TCPDump oder Wireshark → tatsächliche Kommunikation bei Bedarf auf Paketebene analysieren

ARP Scan beantwortet damit sehr schnell eine der grundlegendsten Fragen einer lokalen Netzwerkdiagnose:

Welche IPv4-Geräte sind auf meinem unmittelbar angeschlossenen lokalen Netzwerk tatsächlich per ARP sichtbar?

A.21 MTR

Bereich: Netzwerk
Programm: MTR
Baseline 20261001: MTR 0.95
Installierte Pakete: `mtr 0.95-1.1+b1`, `mtr-tiny 0.95-1.1+b1`
Bedienung: Kommandozeile / interaktive Terminaloberfläche
Aufgabe: Fortlaufende Analyse des Netzwerkpfads mit Hop-, Latenz- und Paketverluststatistiken

Zweck

MTR ist ein Netzwerkdiagnosewerkzeug, das wesentliche Funktionen von:

traceroute

und:

ping

in einer gemeinsamen fortlaufenden Messung verbindet.

Das Programm untersucht den Netzwerkpfad zwischen dem lokalen System und einem Ziel und ermittelt die dazwischenliegenden Router beziehungsweise Hops.

Im Gegensatz zu einem einmaligen klassischen Traceroute führt MTR anschließend fortlaufend weitere Messungen durch.

Dadurch entstehen für jeden sichtbaren Hop statistische Werte über:

Antwortverhalten,

Paketverlust,

Round-Trip-Zeit,

Minimal- und Maximalwerte,

Durchschnittslatenz

und Schwankungen.

Einsatzgebiet im Rescue-System

MTR eignet sich innerhalb von JUMBO BLOCK Rescue insbesondere zur Untersuchung von:

langsamen Netzwerkverbindungen,

sporadischen Verbindungsproblemen,

Paketverlust,

Latenzproblemen,

instabilen WAN-Verbindungen,

Routingpfaden,

Internetverbindungen,

VPN-Pfaden,

Verbindungen zu entfernten Servern,

Unterschieden zwischen IPv4 und IPv6,

und Problemen, die erst nach mehreren Netzwerk-Hops auftreten.

Grundprinzip

MTR sendet Netzwerkprobes mit gezielt begrenzten TTL-Werten.

TTL steht bei IPv4 für:

Time To Live

Bei jedem Router, den ein IP-Paket passiert, wird der TTL-Wert reduziert.

Erreicht der Wert null, wird das Paket nicht weitergeleitet.

Der betreffende Router kann darauf mit einer ICMP-Nachricht antworten.

MTR nutzt diesen Mechanismus, um die einzelnen Hops entlang eines Netzwerkpfads zu ermitteln.

Schrittweise TTL

Vereinfacht beginnt die Untersuchung mit:

TTL 1

danach:

TTL 2

danach:

TTL 3

und so weiter.

Dadurch werden nacheinander die Router entlang des Pfads sichtbar, soweit diese entsprechende Antworten liefern.

Unterschied zu Traceroute

Ein klassischer Traceroute untersucht den Pfad zu einem bestimmten Zeitpunkt.

MTR wiederholt seine Messungen fortlaufend.

Dadurch können Probleme sichtbar werden, die nur zeitweise auftreten.

Beispielsweise:

sporadischer Paketverlust,

schwankende Latenz,

zeitweilige Überlastung,

oder intermittierende Erreichbarkeitsprobleme.

Unterschied zu Ping

Ping untersucht primär die Erreichbarkeit und Round-Trip-Zeit zu einem Ziel.

MTR untersucht dagegen gleichzeitig die sichtbaren Zwischenstationen des Pfads.

Vereinfacht:

Ping → Wie verhält sich das Ziel?

Traceroute → Über welche Hops verläuft der Pfad?

MTR → Wie verhalten sich Ziel und sichtbare Hops fortlaufend?

Grundlegender Aufruf

Ein grundlegender Aufruf lautet:

```
mtr ZIEL
```

Beispielsweise:

```
mtr example.net
```

MTR startet damit normalerweise seine interaktive Terminaldarstellung.

Interaktive Anzeige

In der interaktiven Ansicht aktualisiert MTR die Messwerte fortlaufend.

Dadurch können Veränderungen unmittelbar beobachtet werden.

Ein Hop wird als eigene Zeile dargestellt.

Typische Spalten sind:

```
Loss%
Snt
Last
Avg
Best
Wrst
StDev
```

Loss%

`Loss%` zeigt den Anteil der an diesen Hop gerichteten Probes, für die keine entsprechende Antwort registriert wurde.

Dieser Wert muss sehr sorgfältig interpretiert werden.

Ein hoher Wert an einem Zwischen-Hop beweist nicht automatisch, dass der Router entsprechend viele Transitpakete verwirft.

Snt

`Snt` zeigt die Anzahl der bisher gesendeten Probes.

Eine Statistik mit nur wenigen Probes besitzt naturgemäß weniger Aussagekraft als eine längere Messreihe.

Last

`Last` bezeichnet die Round-Trip-Zeit der jüngsten beantworteten Probe.

Die Einheit ist typischerweise Millisekunden.

Avg

`Avg` ist die durchschnittliche gemessene Round-Trip-Zeit.

Sie hilft, die typische Latenz des jeweiligen Hops über die Messdauer einzuschätzen.

Best

`Best` ist die niedrigste gemessene Round-Trip-Zeit.

Dieser Wert kann einen Hinweis darauf geben, welche Latenz unter günstigen Bedingungen erreichbar war.

Wrst

`Wrst` bezeichnet die höchste gemessene Round-Trip-Zeit.

Ein sehr großer Unterschied zwischen `Best` und `Wrst` weist auf starke Schwankungen in den Antwortzeiten hin.

StDev

`StDev` bezeichnet die Standardabweichung der gemessenen Round-Trip-Zeiten.

Sie beschreibt die Streuung der Messwerte.

Ein größerer Wert bedeutet stärkere Schwankungen.

Latenz

Die von MTR dargestellte Zeit ist eine **Round-Trip-Time**.

Sie beschreibt den Hin- und Rückweg einer Probe.

Sie ist deshalb nicht unmittelbar identisch mit der reinen Laufzeit nur in einer Richtung.

Jitter und Schwankungen

Stark schwankende Round-Trip-Zeiten können für bestimmte Anwendungen problematisch sein.

Dazu gehören beispielsweise:

VoIP,

Videokonferenzen,

interaktive Remote-Sitzungen

und andere zeitkritische Kommunikation.

MTR 0.95 kann neben den klassischen Latenzwerten auch verschiedene Jitter-bezogene Statistikfelder darstellen.

Der wichtigste Interpretationsfehler

Ein typisches MTR-Ergebnis kann beispielsweise so aussehen:

Hop 1: 0 % Verlust
Hop 2: 0 % Verlust
Hop 3: 60 % Verlust
Hop 4: 0 % Verlust
Hop 5: 0 % Verlust
Ziel: 0 % Verlust

Daraus darf **nicht** geschlossen werden:

„Hop 3 verliert 60 % des Netzwerkverkehrs.“

Warum?

Router müssen Transitverkehr weiterleiten und gleichzeitig gegebenenfalls selbst auf Diagnosepakete antworten.

Ein Router kann seine eigenen ICMP-Antworten:

begrenzen,

niedriger priorisieren

oder vollständig unterdrücken,

während er normalen Transitverkehr weiterhin korrekt weiterleitet.

Das offizielle MTR-Projekt weist ausdrücklich auf dieses Verhalten moderner Router hin.

Entscheidendes Interpretationsprinzip

Wenn ein Zwischen-Hop hohen scheinbaren Verlust zeigt, spätere Hops und das Ziel aber keinen entsprechenden Verlust zeigen, spricht dies **nicht für einen nachgewiesenen Transitverlust an diesem Hop**.

Der Router antwortet dann möglicherweise lediglich nicht zuverlässig auf die an ihn selbst gerichteten Diagnoseprobes.

Fortgesetzter Verlust

Anders ist ein Muster wie:

Hop 1: 0 %
Hop 2: 0 %
Hop 3: 20 %
Hop 4: 20 %
Hop 5: 20 %
Ziel: 20 %

Hier beginnt ab Hop 3 ein Verlustmuster, das bis zum Ziel fortbesteht.

Dies ist diagnostisch deutlich relevanter.

Trotzdem muss auch dieses Muster im Gesamtkontext geprüft werden.

Ziel ist besonders wichtig

Für die Bewertung einer Ende-zu-Ende-Verbindung ist der Ziel-Hop von zentraler Bedeutung.

Zeigt das Ziel keinen Paketverlust, dürfen Verluste einzelner Zwischen-Hops nicht automatisch als Ende-zu-Ende-Paketverlust interpretiert werden.

Nicht antwortende Hops

Ein Router muss nicht auf die von MTR verwendeten Diagnoseprobes antworten.

Ein Hop kann deshalb als:

???

oder in einer vergleichbaren Form erscheinen.

Dies bedeutet nicht automatisch, dass dort eine Unterbrechung des Pfads vorliegt.

Wenn spätere Hops antworten, werden die Pakete offensichtlich über beziehungsweise hinter dieser Stelle weitertransportiert.

Firewall und Filter

Firewalls und Router können Diagnoseverkehr unterschiedlich behandeln.

Mögliche Effekte sind:

ICMP wird verworfen,

ICMP wird begrenzt,

UDP wird anders behandelt als ICMP,

TCP-Probes werden anders behandelt,

oder Antworten werden nur teilweise erzeugt.

MTR-Ergebnisse müssen deshalb immer unter Berücksichtigung des verwendeten Probe-Protokolls interpretiert werden.

Namensauflösung

Standardmäßig kann MTR versuchen, IP-Adressen in Hostnamen aufzulösen.

Dies erleichtert die Lesbarkeit, kann aber:

zusätzliche DNS-Kommunikation erzeugen,

die Anzeige verzögern,

oder bei fehlerhaftem DNS störend sein.

DNS deaktivieren

Mit:

`-n`

beziehungsweise:

`--no-dns`

wird die Namensauflösung deaktiviert.

Beispiel:

```
mtr -n ZIEL
```

Damit werden numerische IP-Adressen angezeigt.

Für eine technische Rescue-Diagnose ist dies häufig besonders sinnvoll.

Hostname und IP-Adresse gemeinsam

Mit:

`-b`

beziehungsweise:

`--show-ips`

kann MTR sowohl Hostnamen als auch numerische IP-Adressen anzeigen.

Dies kann bei der Dokumentation eines Netzwerkpfads hilfreich sein.

IPv4 erzwingen

Mit:

`-4`

wird ausschließlich IPv4 verwendet.

Beispiel:

```
mtr -4 ZIEL
```

IPv6 erzwingen

Mit:

`-6`

wird ausschließlich IPv6 verwendet.

Beispiel:

```
mtr -6 ZIEL
```

Damit können IPv4- und IPv6-Pfade getrennt untersucht werden.

Warum IPv4 und IPv6 getrennt testen?

Ein Hostname kann sowohl IPv4- als auch IPv6-Adressen besitzen.

Die beiden Protokollfamilien können:

unterschiedliche Router,

unterschiedliche Providerpfade,

unterschiedliche Firewalls

und unterschiedliche Fehlerbilder

verwenden.

Eine funktionierende IPv4-Verbindung beweist deshalb nicht automatisch, dass auch der IPv6-Pfad funktioniert – und umgekehrt.

Report Mode

Für Dokumentation und reproduzierbare Diagnose ist der Report Mode besonders wichtig.

Er wird aktiviert mit:

`-r`

beziehungsweise:

`--report`

Beispiel:

```
mtr -r ZIEL
```

MTR führt dabei eine definierte Zahl von Messzyklen aus, gibt anschließend einen zusammenfassenden Bericht aus und beendet sich.

Anzahl der Messzyklen

Mit:

```
-c ANZAHL
```

beziehungsweise:

```
--report-cycles ANZAHL
```

wird die Anzahl der Messzyklen festgelegt.

Beispiel:

```
mtr -r -c 100 ZIEL
```

Damit wird ein Bericht nach 100 Messzyklen erzeugt.

Bericht in Datei speichern

Die Textausgabe kann über die Shell in eine Datei geschrieben werden.

Beispiel:

```
mtr -n -r -c 100 ZIEL > mtr-report.txt
```

Damit entsteht ein dokumentierbarer Messbericht.

Warum mehrere Messzyklen?

Ein einzelner Messwert zeigt nur einen kurzen Moment.

Eine größere Zahl von Messungen kann:

sporadischen Verlust,

Latenzspitzen,

zeitliche Schwankungen

und instabile Verbindungen

besser sichtbar machen.

Die Messdauer sollte jedoch dem tatsächlichen Diagnosezweck entsprechen.

Wide Report

Mit:

```
-w
```

beziehungsweise:

`--report-wide`

wird der Wide Report Mode verwendet.

Dabei werden Hostnamen im Bericht nicht auf dieselbe Weise gekürzt.

Dies ist insbesondere bei langen DNS-Namen nützlich.

Beispiel für Rescue-Dokumentation

Ein praxisnaher Aufruf lautet:

```
mtr -n -r -c 100 ZIEL
```

Damit werden:

DNS-Auflösungen vermieden,

100 Messzyklen durchgeführt,

und anschließend die Statistik ausgegeben.

Intervall

Mit:

`-i SEKUNDEN`

beziehungsweise:

`--interval SEKUNDEN`

wird das Intervall zwischen den ICMP-Echo-Anfragen eingestellt.

Der dokumentierte Standardwert beträgt:

1 Sekunde.

Messlast

MTR erzeugt aktiv Netzwerkverkehr.

Die Dokumentation weist ausdrücklich darauf hin, dass jede laufende MTR-Instanz eine relevante Menge Netzwerkverkehr erzeugt und Messungen die Netzwerkleistung beeinflussen können.

In empfindlichen Umgebungen sollten Messdauer und Intervall deshalb bewusst gewählt werden.

Paketgröße

Mit:

`-s PACKETSIZE`

beziehungsweise:

`--psize PACKETSIZE`

wird die Größe der Probe-Pakete festgelegt.

Der Wert umfasst bei ICMP die IP- und ICMP-Header.

Unterschiedliche Paketgrößen können bei bestimmten Problemen unterschiedliche Ergebnisse liefern.

Warum Paketgröße relevant sein kann

Probleme können beispielsweise abhängig sein von:

MTU,

Tunnelverbindungen,

VPNs,

Fragmentierung

oder bestimmten Netzwerkpfaden.

Eine Veränderung der Paketgröße sollte jedoch gezielt erfolgen und nicht als allgemeine Standarddiagnose verstanden werden.

Interface auswählen

Mit:

```
-I NAME
```

beziehungsweise:

```
--interface NAME
```

kann ein bestimmtes Netzwerkinterface für die Probes ausgewählt werden.

Dies ist besonders nützlich, wenn JUMBO BLOCK Rescue mehrere mögliche Netzwerkpfade besitzt, beispielsweise:

Ethernet

und

WLAN.

Quelladresse festlegen

Mit:

```
-a ADDRESS
```

beziehungsweise:

```
--address ADDRESS
```

kann die Ausgangsverbindung an eine bestimmte Quelladresse gebunden werden.

Dies ist bei Systemen mit mehreren IP-Adressen oder komplexeren Routingkonfigurationen hilfreich.

Report-Felder anpassen

Mit:

```
-o FIELDS
```

beziehungsweise:

```
--order FIELDS
```

können die dargestellten Statistikfelder und ihre Reihenfolge festgelegt werden.

MTR unterstützt unter anderem Felder für:

Loss Ratio,

gesendete Pakete,

empfangene Pakete,

aktuelle RTT,

beste RTT,

durchschnittliche RTT,

schlechteste RTT,

Standardabweichung

und verschiedene Jitterwerte.

JSON

MTR 0.95 unterstützt:

```
-j
```

beziehungsweise:

```
--json
```

für JSON-Ausgabe, sofern die dafür erforderliche Bibliothek beim Build verfügbar war.

JSON eignet sich für automatisierte Weiterverarbeitung.

Die Baseline-Paketliste allein belegt jedoch nicht, mit welchen optionalen Build-Funktionen das konkrete Binary erstellt wurde.

Die Verfügbarkeit sollte deshalb auf dem laufenden Rescue-System geprüft werden, bevor sie für einen automatisierten Workflow vorausgesetzt wird.

XML

Mit:

```
-x
```

beziehungsweise:

```
--xml
```

kann XML-Ausgabe erzeugt werden.

Dies ist ebenfalls für maschinelle Weiterverarbeitung vorgesehen.

CSV

MTR unterstützt außerdem eine CSV-Ausgabe.

Damit können Messdaten beispielsweise in Tabellen- und Analysewerkzeuge übernommen werden.

Bei einer späteren automatisierten Verarbeitung sollte die tatsächlich von der installierten Version erzeugte Feldstruktur berücksichtigt werden.

ICMP

Der klassische MTR-Betrieb verwendet ICMP-Echo-Probes.

Dies ist jedoch nicht die einzige unterstützte Methode.

UDP

MTR unterstützt auch UDP-Probes.

Dies kann nützlich sein, wenn ICMP im Netzwerk anders behandelt wird als UDP.

TCP

MTR unterstützt außerdem TCP-Probes.

Dies kann bei der Diagnose eines konkreten TCP-basierten Dienstpfads besonders hilfreich sein.

Zielport

Mit:

```
-P PORT
```

beziehungsweise:

```
--port PORT
```

kann für TCP-, UDP- beziehungsweise unterstützte transportprotokollbasierte Messungen ein Zielport angegeben werden.

Damit kann die Untersuchung stärker an den tatsächlich interessierenden Dienst angepasst werden.

Warum verschiedene Probe-Protokolle wichtig sind

Netzwerke behandeln nicht zwangsläufig:

ICMP,

UDP

und

TCP

gleich.

Firewalls, ACLs, Router und Provider können für diese Protokolle unterschiedliche Regeln besitzen.

Ein ICMP-basierter MTR-Test bildet daher nicht zwingend exakt das Verhalten einer TCP-Anwendung ab.

Beispiel

Wenn eine HTTPS-Verbindung Probleme bereitet, kann eine TCP-basierte Pfaduntersuchung zum relevanten Port zusätzliche Informationen liefern.

Das Ergebnis muss trotzdem vorsichtig interpretiert werden, weil MTR eine Diagnosemessung und keine vollständige Simulation der Anwendung ist.

MPLS

Mit:

`-e`

beziehungsweise:

`--mpls`

kann MTR Informationen aus ICMP-Erweiterungen für MPLS anzeigen, sofern entsprechende Informationen in den Antwortpaketen vorhanden sind.

Dies kann in Provider- oder größeren Unternehmensnetzen zusätzliche Pfadinformationen liefern.

Autonomous-System-Informationen

MTR kann zu Hops auch Informationen über Autonomous Systems anzeigen.

Mit:

`-z`

beziehungsweise:

`--aslookup`

kann die AS-Nummer dargestellt werden.

Solche Informationen erfordern entsprechende externe Zuordnungsdaten beziehungsweise Abfragen.

Sie sind daher nicht mit einer vollständig offline arbeitenden Pfadanalyse gleichzusetzen.

Interaktive Steuerung

Während der interaktiven MTR-Ausführung stehen Tastaturbefehle zur Verfügung.

Dazu gehören unter anderem:

`p` – Messung pausieren

`Space` – fortsetzen

`n` – DNS-Auflösung umschalten

`r` – Statistik zurücksetzen

`j` – zwischen Latenz- und Jitterstatistiken wechseln

`q` – beenden

Weitere Einstellungen können ebenfalls während der Laufzeit angepasst werden.

Statistik zurücksetzen

Das Zurücksetzen der Statistik ist hilfreich, wenn zunächst eine Verbindung vorbereitet und anschließend nur ein bestimmter Zeitraum betrachtet werden soll.

Nach dem Reset beginnen die statistischen Werte erneut.

Route kann sich ändern

Internet- und Unternehmensrouten sind nicht zwingend statisch.

Zwischen zwei Messungen können sich Hops verändern.

Auch während einer laufenden Untersuchung können unterschiedliche Pfade auftreten.

Mögliche Ursachen sind beispielsweise:

Routingänderungen,

Load Balancing,

ECMP,

Provideränderungen

oder Failover.

Load Balancing

Mehrere Netzwerkpfade können dazu führen, dass Diagnosepakete unterschiedliche Router passieren.

Dadurch können MTR-Ergebnisse komplexer aussehen als ein einfacher linearer Pfad.

Ein wechselnder Hop bedeutet deshalb nicht automatisch eine Fehlfunktion.

Asymmetrisches Routing

MTR misst Round-Trip-Zeiten.

Hin- und Rückweg müssen nicht identisch sein.

Ein hoher RTT-Wert an einem Hop lässt deshalb nicht ohne Weiteres erkennen, ob die Verzögerung auf dem Hinweg, Rückweg oder in der Antwortbehandlung des Routers entstanden ist.

ICMP-Priorisierung

Ein Router kann Transitpakete mit hoher Priorität weiterleiten, während Antworten an seine eigene Control Plane gedrosselt werden.

Genau deshalb muss Zwischen-Hop-Verlust vorsichtig bewertet werden.

Beispiel: scheinbarer Verlust an einem Zwischen-Hop

Messung:

Hop 1: 0 %
Hop 2: 40 %
Hop 3: 0 %
Hop 4: 0 %
Ziel: 0 %

Interpretation:

Der 40-%-Wert an Hop 2 beweist keinen 40-%-Verlust des tatsächlichen Ende-zu-Ende-Verkehrs.

Wenn Hop 2 tatsächlich 40 % aller Transitpakete verwürfe, müssten entsprechende Auswirkungen normalerweise auch an den nachfolgenden Hops sichtbar werden.

Eine plausible Erklärung ist daher eine begrenzte Antwortbereitschaft des Routers auf die Diagnoseprobes.

Beispiel: Verlust setzt sich fort

Hop 1: 0 %
Hop 2: 0 %
Hop 3: 15 %
Hop 4: 15 %
Hop 5: 15 %
Ziel: 15 %

Dieses Muster ist deutlich verdächtiger.

Ab Hop 3 ist ein vergleichbarer Verlust bis zum Ziel sichtbar.

Hop 3 beziehungsweise die Verbindung in diesem Bereich wird dadurch zu einem wichtigen Untersuchungspunkt.

Es handelt sich dennoch um eine Diagnoseindikation und nicht automatisch um einen endgültigen Beweis für die genaue physische Fehlerstelle.

Beispiel: Latenzsprung

Hop 1: 1 ms
Hop 2: 2 ms
Hop 3: 3 ms
Hop 4: 35 ms
Hop 5: 36 ms
Ziel: 36 ms

Ein dauerhafter Latenzanstieg ab Hop 4 kann auf einen längeren Netzwerkabschnitt oder eine andere Pfadcharakteristik hinweisen.

Ein solcher Sprung ist nicht automatisch ein Fehler.

Bei WAN- und Internetpfaden kann beispielsweise eine große geografische Distanz hinzukommen.

Geografische Distanz

Round-Trip-Zeit wird physikalisch unter anderem durch die Länge des Übertragungswegs begrenzt.

Eine Verbindung über große geografische Entfernungen besitzt deshalb zwangsläufig eine höhere Mindestlatenz als eine lokale Verbindung.

MTR kann diese Latenz sichtbar machen, aber nicht allein aus ihr auf einen Defekt schließen.

Beispiel: starke Schwankungen

Best: 10 ms
Avg: 25 ms
Wrst: 180 ms
StDev: hoch

Dies zeigt eine stark schwankende Antwortzeit.

Mögliche Ursachen müssen anschließend weiter untersucht werden.

MTR allein identifiziert nicht automatisch die Ursache.

Typischer Rescue-Ablauf

Ein kontrollierter MTR-Einsatz kann folgendermaßen erfolgen:

1. Ziel festlegen

Das tatsächlich problematische Ziel verwenden.

2. Protokollfamilie berücksichtigen

Falls erforderlich IPv4 und IPv6 getrennt testen.

3. DNS gegebenenfalls deaktivieren

Beispiel:

```
mtr -n ZIEL
```

4. Messung ausreichend lange laufen lassen

Nicht aus einzelnen frühen Messwerten sofort Schlussfolgerungen ziehen.

5. Ziel-Hop zuerst bewerten

Ist dort Paketverlust oder ungewöhnliche Latenz vorhanden?

6. Danach Zwischen-Hops vergleichen

Wo beginnt ein Muster, das sich bis zum Ziel fortsetzt?

7. Einzelne isolierte Verlustwerte nicht überinterpretieren

8. Bei Bedarf Report erzeugen

Beispiel:

```
mtr -n -r -c 100 ZIEL
```

9. Bei Protokollverdacht alternative Probe-Methode testen

ICMP, UDP oder TCP entsprechend dem Diagnosefall.

10. Bei Bedarf mit TCPDump oder Wireshark ergänzen

Damit kann der tatsächliche Probe- und Antwortverkehr auf Paketebene untersucht werden.

MTR und Nmap

Nmap untersucht:

Hosts,

Ports,

Dienste

und weitere Systemeigenschaften.

MTR untersucht:

den Netzwerkpfad

und dessen fortlaufendes Antwortverhalten.

Ein Server kann mit Nmap erreichbar sein, während MTR gleichzeitig ungewöhnliche Latenz oder Paketverluste entlang des Pfads sichtbar macht.

MTR und Wireshark

Wireshark kann die von MTR erzeugten Netzwerkpakete aufzeichnen.

Dadurch lässt sich genau untersuchen:

welche Probes gesendet werden,

welche ICMP-Antworten zurückkommen,

und welche Netzwerkpakete tatsächlich am lokalen Capture-Punkt sichtbar sind.

MTR und TCPDump

TCPDump kann MTR-Verkehr terminalbasiert mitschneiden.

Dies ist hilfreich, wenn:

MTR ungewöhnliche Ergebnisse liefert,

Antwortpakete genauer untersucht werden sollen,

oder dokumentiert werden muss, welche Pakete tatsächlich übertragen wurden.

MTR und ARP Scan

ARP Scan untersucht das unmittelbar lokale IPv4-Layer-2-Netz.

MTR untersucht dagegen einen gerouteten IP-Pfad über mehrere Hops.

Damit ergänzen sich beide Werkzeuge:

ARP Scan → Wer befindet sich lokal im Netz?

MTR → Über welchen IP-Pfad erreiche ich ein entferntes Ziel und wie verhält sich dieser Pfad?

MTR und Dig

Dig untersucht gezielt DNS.

MTR untersucht den Netzwerkpfad.

Wenn beispielsweise ein DNS-Server nur langsam erreichbar ist, kann:

Dig die DNS-Antwort selbst untersuchen

und:

MTR den Netzwerkpfad zum DNS-Server analysieren.

Grenzen

MTR zeigt nur Router, die auf die verwendeten Diagnoseprobes entsprechend reagieren.

Nicht antwortende Zwischen-Hops bedeuten nicht automatisch einen unterbrochenen Pfad.

Paketverlust an einem Zwischen-Hop ist nicht automatisch Transit-Paketverlust.

Round-Trip-Zeit zeigt nicht getrennt die Latenz von Hin- und Rückweg.

Routing kann sich während oder zwischen Messungen ändern.

Load Balancing kann unterschiedliche Pfade erzeugen.

ICMP-, UDP- und TCP-Probes können von Netzwerkkomponenten unterschiedlich behandelt werden.

MTR identifiziert nicht automatisch die physische Ursache eines Problems.

Eine fehlerfreie MTR-Messung beweist nicht, dass jede Anwendung und jedes Protokoll fehlerfrei funktioniert.

Sicherheitshinweise

MTR nur gegen Systeme und Netze einsetzen, deren Diagnose zulässig ist.

Messungen erzeugen aktiv Netzwerkverkehr.

Messdauer und Probe-Intervall in empfindlichen Netzen bewusst wählen.

Zwischen-Hop-Verlust niemals isoliert als Beweis eines Netzwerkfehlers interpretieren.

Ziel-Hop und nachfolgende Hops in die Bewertung einbeziehen.

IPv4 und IPv6 bei Dual-Stack-Problemen gegebenenfalls getrennt untersuchen.

Bei servicebezogenen Problemen berücksichtigen, dass ICMP anders behandelt werden kann als TCP oder UDP.

Berichte können interne IP-Adressen, Hostnamen und Routinginformationen enthalten und sind entsprechend zu schützen.

Technische Einordnung in JUMBO BLOCK Rescue

MTR übernimmt innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue die Rolle der **fortlaufenden Pfad-, Latenz- und Paketverlustanalyse**.

Damit ergibt sich bisher:

A.17 Nmap

aktive Host-, Port- und Dienstanalyse.

A.18 Wireshark

grafische Paket- und Protokollanalyse.

A.19 TCPDump

terminalbasierte Paketaufzeichnung und Filterung.

A.20 ARP Scan

lokale IPv4-Geräteerkennung auf Layer 2.

A.21 MTR

fortlaufende Analyse des gerouteten Netzwerkpfads.

Ein besonders wichtiger Diagnoseablauf lautet:

ARP Scan prüft die lokale Umgebung.

MTR verfolgt den Pfad zum entfernten Ziel.

Nmap untersucht das Ziel und seine Dienste.

TCPDump oder Wireshark untersuchen bei Bedarf die tatsächlich übertragenen Pakete.

MTR schließt damit die diagnostische Lücke zwischen einer einfachen lokalen Erreichbarkeitsprüfung und einer detaillierten Paketaufzeichnung.

Seine größte Stärke liegt nicht allein darin, einen Netzwerkpfad anzuzeigen, sondern dessen Verhalten **über eine Messreihe hinweg** sichtbar zu machen.

Für die korrekte Anwendung gilt dabei als zentrale Regel:

Ein auffälliger Zwischen-Hop ist erst dann besonders aussagekräftig, wenn sich das auffällige Verhalten in den nachfolgenden Hops bis zum Ziel fortsetzt.

A.22 Dig

Bereich: Netzwerk
Programm: Dig
Technische Basis: BIND 9 DNS Utilities
Baseline 20261001: BIND 9.20.29
Installiertes Paket: `bind9-dnsutils 1:9.20.29-1~deb13u1`
Bedienung: Kommandozeile
Aufgabe: Direkte DNS-Abfragen, Resolverdiagnose, Untersuchung von Resource Records und DNS-Delegationen

Zweck

Dig steht für:

Domain Information Groper

und ist ein flexibles Kommandozeilenwerkzeug zur Abfrage von DNS-Nameservern.

Dig sendet gezielte DNS-Abfragen und zeigt die vom angesprochenen Nameserver zurückgelieferten DNS-Antworten detailliert an.

Damit kann unabhängig von Browsern oder anderen Anwendungen untersucht werden:

welcher DNS-Server antwortet,

welche Resource Records vorhanden sind,

welche IPv4- oder IPv6-Adressen zurückgegeben werden,

welche Nameserver für eine Domain zuständig sind,

welche Mailserver eingetragen sind,

welche TTL-Werte gelten,

welcher DNS-Statuscode zurückgegeben wird,

ob ein bestimmter Resolver anders antwortet als ein anderer,

und wie eine DNS-Delegation von der Root-Zone bis zum Ziel verläuft.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich Dig insbesondere für:

DNS-Fehlerdiagnose,

Überprüfung der konfigurierten DNS-Resolver,

direkte Abfrage bestimmter DNS-Server,

Vergleich mehrerer Resolver,

A- und AAAA-Abfragen,

MX-, NS-, TXT-, CNAME- und SOA-Abfragen,

Reverse DNS,

Analyse von DNS-Delegationen,

Untersuchung von DNSSEC-bezogenen Daten,

Vergleich von UDP- und TCP-DNS,

und die Trennung eines DNS-Problems von einem allgemeinen Netzwerkproblem.

DNS ist ein eigenes Diagnosesystem

Ein häufiger Diagnosefehler besteht darin, fehlgeschlagene Namensauflösung unmittelbar mit fehlender Netzwerkverbindung gleichzusetzen.

Beispiel:

`server.example` lässt sich nicht auflösen.

Daraus folgt nicht automatisch:

„Das Netzwerk funktioniert nicht.“

Mögliche Ursachen sind beispielsweise:

DNS-Server nicht erreichbar,

falscher DNS-Server konfiguriert,

Name existiert nicht,

Record fehlt,

Delegation fehlerhaft,

Resolver liefert einen Fehler,

oder DNS-Kommunikation wird gefiltert.

Dig ermöglicht es, diese Ebene gezielt zu untersuchen.

Grundsyntax

Die typische Syntax lautet:

```
dig [@server] name [type]
```

Beispiel:

```
dig example.com
```

Wird kein Record-Typ angegeben, führt Dig standardmäßig eine Abfrage nach einem:

A

Record durch.

Standardresolver

Wird kein bestimmter DNS-Server mit `@server` angegeben, verwendet Dig die in:

```
/etc/resolv.conf
```

eingetragenen Nameserver.

Damit lässt sich zunächst genau der DNS-Pfad untersuchen, den das Rescue-System aufgrund seiner Resolverkonfiguration verwendet.

Kein verwendbarer Resolver eingetragen

Findet Dig dort keine verwendbare Serveradresse, versucht es laut BIND-Dokumentation die Anfrage an das lokale System zu senden.

Deshalb ist es für eine Diagnose wichtig, nicht allein anzunehmen, welcher DNS-Server verwendet wird, sondern die Ausgabe und Systemkonfiguration tatsächlich zu prüfen.

Bestimmten DNS-Server abfragen

Ein DNS-Server kann ausdrücklich angegeben werden:

```
dig @SERVER example.com
```

SERVER kann dabei eine IPv4- oder IPv6-Adresse oder ein auflösbarer Hostname sein.

Beispielprinzip:

```
dig @192.0.2.53 example.com
```

Die verwendete IP-Adresse dient ausschließlich als Dokumentationsbeispiel.

Warum `@server` wichtig ist

Damit lassen sich verschiedene Resolver direkt miteinander vergleichen.

Beispielsweise:

lokaler DNS-Server,

Unternehmensresolver,

VPN-DNS-Server,

autoritativer Nameserver,

oder ein anderer ausdrücklich vorgesehener Resolver.

Wenn zwei Server unterschiedliche Ergebnisse liefern, ist dies ein wichtiger Diagnosehinweis.

Direkte Serverabfrage und Systemresolver unterscheiden

Ein:

```
dig example.com
```

und ein:

```
dig @SERVER example.com
```

müssen diagnostisch getrennt betrachtet werden.

Der erste Aufruf verwendet die Resolverkonfiguration des Systems.

Der zweite richtet die DNS-Anfrage ausdrücklich an den angegebenen Server.

Dadurch kann festgestellt werden, ob das Problem möglicherweise an der lokalen Resolverauswahl liegt.

Aufbau der Standardausgabe

Eine normale Dig-Ausgabe enthält mehrere Bereiche.

Dazu gehören typischerweise:

Header,

QUESTION SECTION,

ANSWER SECTION,

AUTHORITY SECTION,

ADDITIONAL SECTION

und abschließende Abfrageinformationen.

Nicht jede Antwort enthält alle Bereiche.

Header

Der Header enthält zentrale Informationen über die DNS-Antwort.

Dazu gehören insbesondere:

Opcode,

Status,

Flags,

Query-ID

und die Anzahl der Records in den verschiedenen Abschnitten.

Status

Besonders wichtig ist:

`status:`

Ein erfolgreicher DNS-Transport bedeutet nicht automatisch, dass der gesuchte Name existiert.

Beispielsweise kann Dig eine ordnungsgemäße DNS-Antwort erhalten, deren Status:

`NXDOMAIN`

lautet.

NOERROR

`NOERROR`

bedeutet, dass die DNS-Anfrage ohne DNS-Protokollfehler verarbeitet wurde.

Das bedeutet jedoch nicht zwingend, dass für den angefragten Record-Typ auch ein Ergebnis in der Answer Section vorhanden sein muss.

NXDOMAIN

`NXDOMAIN`

bedeutet, dass der angefragte DNS-Name nicht existiert.

Dies ist eine gültige DNS-Antwort.

Ein sehr wichtiges technisches Detail der BIND-Dokumentation lautet deshalb:

Dig beendet sich auch bei einer empfangenen **NXDOMAIN-Antwort mit Exitcode 0**.

Für Skripte darf daher:

Exitcode 0

nicht mit:

Name existiert

gleichgesetzt werden.

Keine Antwort vom Server

Wenn Dig keine Antwort vom DNS-Server erhält, unterscheidet sich dies grundsätzlich von NXDOMAIN.

Dann liegt beispielsweise ein Problem mit:

Erreichbarkeit,

Firewall,

Routing,

DNS-Dienst,

Transport

oder Serverantwort

nahe.

QUESTION SECTION

Die Question Section zeigt, was tatsächlich abgefragt wurde.

Typischerweise enthält sie:

Name,

DNS-Klasse

und Record-Typ.

Die übliche DNS-Klasse für Internet-DNS lautet:

`IN`.

ANSWER SECTION

Die Answer Section enthält die eigentlichen Resource Records, die als Antwort auf die Anfrage geliefert wurden.

Ein Eintrag enthält typischerweise:

Name,

TTL,

Klasse,

Record-Typ

und Record-Daten.

TTL

TTL steht für:

Time To Live

und beschreibt, wie lange ein DNS-Record entsprechend den DNS-Regeln zwischengespeichert werden kann.

Dig zeigt TTL-Werte standardmäßig numerisch an.

Mit:

`+ttlunits`

können TTL-Werte in besser lesbaren Zeiteinheiten wie Sekunden, Minuten, Stunden, Tagen oder Wochen dargestellt werden.

AUTHORITY SECTION

Die Authority Section kann Informationen über die für den betreffenden DNS-Bereich zuständigen Nameserver beziehungsweise Autoritätsinformationen enthalten.

Ihr Inhalt hängt von der konkreten Antwort ab.

ADDITIONAL SECTION

Die Additional Section kann zusätzliche Resource Records enthalten, die für die Antwort hilfreich sind.

Moderne DNS-Antworten können hier beziehungsweise in der Ausgabe außerdem EDNS-bezogene Informationen enthalten.

Query Time

Dig zeigt normalerweise die gemessene Abfragezeit an.

Diese Zeit beschreibt die DNS-Abfrage aus Sicht des Dig-Prozesses.

Sie darf nicht automatisch als reine Netzwerklatenz interpretiert werden.

Bei einem rekursiven Resolver können beispielsweise auch:

Cache-Zustand,

rekursive Auflösung,

DNSSEC-Verarbeitung

und Serverauslastung

die Antwortzeit beeinflussen.

SERVER

Am Ende der Standardausgabe zeigt Dig, welcher DNS-Server die Antwort geliefert hat.

Dies ist bei Rescue-Diagnosen besonders wichtig.

Damit lässt sich kontrollieren, ob tatsächlich der erwartete Resolver geantwortet hat.

MSG SIZE

Dig zeigt außerdem die Größe der empfangenen DNS-Nachricht an.

Dies kann unter anderem bei Untersuchungen zu großen DNS-Antworten, EDNS oder Transportproblemen relevant sein.

A Record

Ein:

A

Record ordnet einem Namen eine IPv4-Adresse zu.

Beispiel:

```
dig example.com A
```

Da A der Standardtyp ist, genügt auch:

```
dig example.com
```

AAAA Record

Ein:

AAAA

Record enthält eine IPv6-Adresse.

Beispiel:

```
dig example.com AAAA
```

Damit kann unabhängig geprüft werden, ob für einen Namen IPv6-Daten veröffentlicht sind.

A und AAAA getrennt untersuchen

Bei Dual-Stack-Systemen sollten beide Record-Typen gegebenenfalls separat geprüft werden.

Eine funktionierende A-Auflösung beweist nicht, dass ein AAAA-Record korrekt vorhanden ist – und umgekehrt.

Dies ergänzt die getrennte IPv4-/IPv6-Pfadanalyse mit MTR.

CNAME

Ein:

CNAME

Record definiert einen kanonischen Namen für einen Alias.

Abfrage:

```
dig example.com CNAME
```

CNAME-Ketten können bei der Fehlersuche relevant sein, weil der ursprünglich verwendete Name auf einen anderen DNS-Namen verweist.

MX

`MX`

steht für Mail Exchange.

Abfrage:

```
dig example.com MX
```

MX-Records bestimmen die für eine Domain veröffentlichten Mailserver und enthalten Prioritätswerte.

Ein MX-Eintrag beweist jedoch nicht, dass der betreffende Mailserver aktuell erreichbar oder funktionsfähig ist.

Er dokumentiert zunächst die DNS-Konfiguration.

NS

Mit:

`NS`

werden Nameserver-Records abgefragt.

Beispiel:

```
dig example.com NS
```

Damit kann untersucht werden, welche Nameserver für die betreffende DNS-Zone veröffentlicht sind.

SOA

`SOA`

steht für:

Start of Authority

Abfrage:

```
dig example.com SOA
```

Der SOA-Record enthält zentrale Verwaltungsinformationen einer DNS-Zone.

Dazu gehören unter anderem:

primärer Nameserver,

verantwortliche Kontaktinformation in DNS-Darstellung,

Seriennummer

und verschiedene Zeitparameter.

Seriennummer

Die SOA-Seriennummer ist insbesondere bei der Untersuchung mehrerer autoritativer Nameserver hilfreich.

Unterschiedliche Seriennummern können darauf hinweisen, dass Server nicht denselben Zonenstand ausliefern.

Die genaue Bewertung hängt jedoch von der verwendeten DNS-Infrastruktur ab.

TXT

TXT-Records enthalten Textdaten.

Abfrage:

```
dig example.com TXT
```

TXT wird für unterschiedliche Zwecke eingesetzt.

Dazu können beispielsweise Domain-Verifikationen und verschiedene Mail-bezogene Richtlinien gehören.

Dig interpretiert den geschäftlichen Zweck eines TXT-Eintrags nicht automatisch.

Es zeigt die veröffentlichten DNS-Daten.

CAA

CAA-Records können ebenfalls gezielt abgefragt werden:

```
dig example.com CAA
```

Sie enthalten DNS-basierte Angaben im Zusammenhang mit der Ausstellung von Zertifikaten für eine Domain.

Dig prüft dadurch jedoch kein vorhandenes TLS-Zertifikat.

Es fragt lediglich den betreffenden DNS-Record ab.

SRV

SRV-Records können Dienstinformationen mit:

```
dig NAME SRV
```

abgefragt werden.

Sie können unter anderem:

Priorität,

Gewichtung,

Port

und Zielhost

enthalten.

PTR und Reverse DNS

Mit:

`-x`

führt Dig eine vereinfachte Reverse-DNS-Abfrage durch.

Beispiel:

```
dig -x 192.0.2.1
```

Dig erzeugt daraus automatisch die passende Reverse-DNS-Anfrage und verwendet:

`PTR`

als Record-Typ.

IPv4 Reverse DNS

Für IPv4 werden Reverse-DNS-Daten unter:

`in-addr.arpa`

abgebildet.

Der Benutzer muss die entsprechende umgekehrte DNS-Schreibweise bei Verwendung von `-x` nicht selbst erzeugen.

IPv6 Reverse DNS

`dig -x`

unterstützt ebenfalls IPv6-Adressen.

Dig erzeugt hierfür die entsprechende Nibble-Darstellung unter:

`IP6.ARPA`.

Reverse DNS ist nicht Forward DNS

Eine IPv4-Adresse kann einen PTR-Record besitzen, der auf einen Namen zeigt.

Das bedeutet nicht automatisch, dass dieser Name wiederum über A oder AAAA auf genau dieselbe Adresse zurückverweist.

Forward- und Reverse-DNS sind getrennte DNS-Datensätze und sollten bei Bedarf separat geprüft werden.

Kurzausgabe

Mit:

`+short`

wird eine stark reduzierte Ausgabe erzeugt.

Beispiel:

```
dig +short example.com
```

Die offizielle Dokumentation beschreibt `+short` als Umschaltung von ausführlicher auf knappe Ausgabe.

Wann `+short` sinnvoll ist

`+short` eignet sich besonders für:

schnelle Kontrollen,

Shell-Skripte,

Weiterverarbeitung

und kompakte Ausgabe.

Für eine vollständige DNS-Fehleranalyse sollte jedoch häufig zunächst die normale Ausgabe verwendet werden, weil dort Status, Flags und Serverinformationen sichtbar sind.

Nur Answer Section

Eine weitere nützliche Form ist:

```
dig +noall +answer example.com
```

Damit wird die Ausgabe gezielt auf die Antwortrecords reduziert.

Dies ist kompakter als die Standardausgabe, erhält aber mehr DNS-Struktur als eine reine `+short`-Ausgabe.

IPv4-Transport erzwingen

Mit:

`-4`

verwendet Dig ausschließlich IPv4 für den Transport zum DNS-Server.

Beispiel:

```
dig -4 example.com
```

Dies bestimmt den **Transport zum DNS-Server**.

Es bestimmt nicht automatisch den abgefragten Record-Typ.

IPv6-Transport erzwingen

Mit:

`-6`

wird ausschließlich IPv6 für den Transport der DNS-Anfrage verwendet.

Beispiel:

```
dig -6 example.com
```

Auch hier muss zwischen:

Transportprotokoll zum Resolver

und:

abgefragtem DNS-Record-Typ

unterschieden werden.

Wichtige Unterscheidung

```
dig -6 example.com A
```

bedeutet:

DNS-Server über IPv6 kontaktieren,

aber einen IPv4-A-Record abfragen.

```
dig -4 example.com AAAA
```

bedeutet:

DNS-Server über IPv4 kontaktieren,

aber einen IPv6-AAAA-Record abfragen.

Diese Trennung ist für Dual-Stack-Diagnosen sehr nützlich.

UDP und TCP

Normale DNS-Abfragen verwendet Dig standardmäßig über UDP, sofern nicht ein Abfragetyp oder eine Option einen anderen Transport erfordert.

Mit:

`+tcp`

kann TCP ausdrücklich erzwungen werden.

Beispiel:

```
dig +tcp example.com
```

Die BIND-9.20-Dokumentation bestätigt außerdem, dass Dig nach einer über UDP als abgeschnitten markierten Antwort normalerweise auf TCP wechseln kann.

Warum TCP testen?

Wenn:

```
dig example.com
```

scheitert, aber:

```
dig +tcp example.com
```

funktioniert,

ist dies ein wichtiger Diagnosehinweis.

Dann sollten insbesondere Unterschiede bei:

Firewall,

UDP-Transport,

Fragmentierung,

Netzwerkpfad

oder DNS-Infrastruktur

weiter untersucht werden.

DNS verwendet nicht nur UDP

Die pauschale Aussage:

„DNS läuft über UDP Port 53“

ist unvollständig.

Klassisches DNS verwendet sowohl UDP als auch TCP.

Darüber hinaus unterstützt die hier dokumentierte Dig-Version auch weitere DNS-Transporte.

DNS over TLS

BIND 9.20 unterstützt in Dig:

```
+tls
```

für DNS over TLS.

Dabei wird standardmäßig Port:

```
853
```

verwendet.

Für die Zertifikatsprüfung stehen weitere TLS-bezogene Optionen zur Verfügung.

Nichtstandard-Port

Mit:

`-p PORT`

kann ein anderer Zielport für die DNS-Abfrage verwendet werden.

Beispielprinzip:

```
dig @SERVER -p PORT example.com
```

Dies ist für DNS-Server nützlich, die zu Test- oder Diagnosezwecken auf einem nicht standardmäßigen Port lauschen.

Timeout

Mit:

`+timeout=T`

wird der Timeout einer Abfrage in Sekunden festgelegt.

Der dokumentierte Standardwert von BIND 9.20 beträgt:

5 Sekunden.

Wiederholungsversuche

Mit:

`+tries=T`

kann die Anzahl der Abfrageversuche beeinflusst werden.

Der dokumentierte Standardwert beträgt:

3 Versuche.

Timeout- und Retry-Werte sollten bei der Diagnose bewusst verändert werden, weil sehr kurze Werte langsame, aber funktionierende DNS-Dienste fälschlich wie einen Ausfall erscheinen lassen können.

Rekursion

Normale Dig-Abfragen setzen standardmäßig das:

RD – Recursion Desired

Bit.

Damit wird ein rekursiver Resolver gebeten, die vollständige Namensauflösung für den Client durchzuführen.

Mit:

`+norecurse`

kann dieses Verhalten deaktiviert werden.

Rekursiver Resolver

Ein rekursiver Resolver übernimmt für den Client die notwendigen weiteren DNS-Abfragen.

Das ist der typische Betriebsmodus für DNS-Server, die Endgeräten als Resolver zur Verfügung gestellt werden.

Autoritativer Nameserver

Ein autoritativer Nameserver ist dagegen für die DNS-Daten einer bestimmten Zone zuständig.

Mit Dig kann ein solcher Server gezielt direkt angesprochen werden:

```
dig @AUTHORITATIVE-SERVER example.com A
```

Damit lässt sich vergleichen:

Was liefert der autoritative Server?

Was liefert der verwendete rekursive Resolver?

Cache-Unterschiede

Ein rekursiver Resolver kann Daten aus seinem Cache liefern.

Ein autoritativer Server liefert dagegen die für seine Zone autoritativen Daten.

Dadurch können während Änderungen vorübergehend unterschiedliche Ergebnisse sichtbar sein.

TTL-Werte spielen dabei eine wichtige Rolle.

`+trace`

Eine der leistungsfähigsten Diagnosefunktionen von Dig ist:

```
+trace
```

Beispiel:

```
dig +trace example.com
```

Dig verfolgt dabei den Delegationspfad ausgehend von den Root-Nameservern.

Es führt iterative Abfragen durch und folgt den erhaltenen Referrals bis zur Auflösung des gesuchten Namens.

Was `+trace` zeigt

Vereinfacht kann sichtbar werden:

Root-Zone `.`

→ zuständige Top-Level-Domain

→ Nameserver der Zielzone

→ Antwort der Zielzone.

Damit kann festgestellt werden, an welcher Stelle einer DNS-Delegationskette Probleme auftreten.

`+trace` ist keine normale Resolverabfrage

Ein:

```
dig example.com
```

fragt normalerweise den konfigurierten rekursiven Resolver.

Ein:

```
dig +trace example.com
```

führt dagegen selbst iterative Abfragen entlang der Delegationskette durch.

Die beiden Tests untersuchen daher unterschiedliche DNS-Pfade.

`@server` zusammen mit `+trace`

Wird bei `+trace` zusätzlich `@server` angegeben, betrifft dieser Server laut BIND-Dokumentation nur die anfängliche Abfrage nach den Root-Nameservern.

Die anschließende Delegationsverfolgung erfolgt entsprechend den erhaltenen Referrals.

DNSSEC-Daten

Mit:

```
+dnssec
```

fordert Dig DNSSEC-bezogene Daten an.

Dies setzt insbesondere das DO-Bit für die DNS-Abfrage.

Dadurch können beispielsweise zu signierten Records passende RRSIG-Daten sichtbar werden.

`+dnssec` bedeutet nicht automatisch vollständige Validierung

Dig ist in erster Linie ein DNS-Abfrage- und Diagnosewerkzeug.

Das bloße Vorhandensein einer RRSIG in der Ausgabe beweist nicht allein, dass eine vollständige DNSSEC-Vertrauenskette erfolgreich validiert wurde.

Für DNSSEC-spezifische Validierungsaufgaben existieren innerhalb der BIND-Werkzeugfamilie zusätzliche Werkzeuge wie `delv`.

`+trace` und DNSSEC

BIND 9.20 aktiviert bei `+trace` auch `+dnssec`, um die Abfragen stärker dem Verhalten eines Nameservers anzunähern.

DNS-Flags

In der Dig-Ausgabe können verschiedene DNS-Flags erscheinen.

Wichtige Beispiele sind:

`qr` – Query Response

`aa` – Authoritative Answer

`rd` – Recursion Desired

`ra` – Recursion Available

`ad` – Authentic Data

Die Bedeutung eines Flags muss immer im Kontext der konkreten Antwort betrachtet werden.

aa

Das Flag:

`aa`

zeigt an, dass die Antwort als autoritative Antwort gekennzeichnet ist.

Dies ist bei der direkten Abfrage eines autoritativen Servers besonders relevant.

ra

`ra`

steht für:

Recursion Available

und zeigt an, dass der antwortende Server Rekursion anbietet.

ad

`ad`

steht für:

Authentic Data.

Bei Antworten eines validierenden Resolvers kann dieses Flag anzeigen, dass der Resolver die betreffenden Daten entsprechend DNSSEC validiert hat.

Das Flag ist dabei eine Aussage des antwortenden Resolvers.

Mehrere Abfragen

Die BIND-Implementierung von Dig kann mehrere DNS-Abfragen in einem Aufruf ausführen.

Zusätzlich unterstützt Dig einen Batch-Modus.

Batch-Datei

Mit:

`-f DATEI`

können Abfragen aus einer Datei gelesen werden.

Dies ist für wiederholbare Diagnoseabläufe nützlich.

Benutzerkonfiguration `.digrc`

Dig kann benutzerspezifische Standardoptionen aus:

`${HOME}/.digrc`

einlesen.

Dadurch kann sich das Verhalten eines Dig-Aufrufs verändern, ohne dass die betreffende Option unmittelbar auf der Kommandozeile sichtbar ist.

Reproduzierbare Diagnose

Mit:

`-r`

wird das Einlesen von:

`${HOME}/.digrc`

unterdrückt.

Die BIND-Dokumentation empfiehlt diese Möglichkeit insbesondere für Skripte, die ein vorhersagbares Verhalten benötigen.

Für dokumentierte Rescue-Abläufe kann dies ebenfalls sinnvoll sein, wenn benutzerspezifische Dig-Optionen ausgeschlossen werden sollen.

Exitcodes

Dig verwendet definierte Exitcodes.

Wichtige Werte sind:

`0` – DNS-Antwort empfangen, ausdrücklich auch bei NXDOMAIN

`1` – Bedienungs-/Syntaxfehler

`8` – Batch-Datei konnte nicht geöffnet werden

`9` – keine Antwort vom Server

`10` – interner Fehler.

Bedeutung für Skripte

Ein Skript darf deshalb nicht nur prüfen:

`dig` liefert Exitcode 0.

Es muss zusätzlich den tatsächlichen DNS-Status beziehungsweise die erwarteten Records auswerten.

Sonst könnte eine NXDOMAIN-Antwort fälschlich als erfolgreiche Namensauflösung interpretiert werden.

Zone Transfer

Dig unterstützt auch:

AXFR

für vollständige Zonentransfers.

Beispielprinzip:

```
dig @SERVER example.com AXFR
```

AXFR verwendet TCP.

AXFR ist kein allgemeines Discovery-Verfahren

Ein autoritativer DNS-Server kann Zonentransfers auf ausdrücklich berechtigte Systeme beschränken.

Ein verweigerter AXFR-Versuch bedeutet daher nicht, dass DNS fehlerhaft arbeitet.

In produktiven DNS-Infrastrukturen ist eine Zugriffsbeschränkung auf Zonentransfers üblich und sicherheitsrelevant.

Zonentransfers dürfen nur in autorisierten Umgebungen durchgeführt werden.

Typischer Rescue-Ablauf

Eine strukturierte DNS-Diagnose kann folgendermaßen erfolgen:

1. Normale Auflösung prüfen

```
dig example.com
```

2. Tatsächlich verwendeten Resolver kontrollieren

SERVER-Zeile der Ausgabe prüfen.

3. A und AAAA getrennt untersuchen

```
dig example.com A
```

```
dig example.com AAAA
```

4. Bei Bedarf einen bestimmten Resolver direkt abfragen

```
dig @SERVER example.com A
```

5. Ergebnisse mehrerer Resolver vergleichen

6. Autoritative Nameserver bestimmen

```
dig example.com NS
```

7. Autoritativen Server direkt abfragen

8. Bei Delegationsproblemen `+trace` einsetzen

```
dig +trace example.com
```

9. Bei Transportverdacht TCP testen

```
dig +tcp example.com
```

10. Bei Bedarf Netzwerkverkehr parallel mit TCPDump oder Wireshark beobachten

Beispiel: Browser meldet „Server nicht gefunden“

Zuerst:

```
dig example.com
```

liefert Dig eine gültige Adresse, funktioniert die grundlegende DNS-Abfrage aus Sicht von Dig.

Dann muss weiter geprüft werden, ob das Problem beispielsweise:

im Browser,

in einer Proxykonfiguration,

bei einer anderen Resolverbibliothek,

bei IPv4/IPv6

oder auf Anwendungsebene

liegt.

Beispiel: Dig erhält keine Antwort

Wenn:

```
dig example.com
```

keine DNS-Antwort erhält, kann gezielt ein anderer vorgesehener DNS-Server getestet werden:

```
dig @SERVER example.com
```

Funktioniert die direkte Abfrage, liegt ein wichtiger Hinweis auf ein Problem mit der normalen Resolverkonfiguration oder dem normalerweise verwendeten DNS-Server vor.

Beispiel: A funktioniert, AAAA nicht

```
dig example.com A
```

liefert eine IPv4-Adresse.

```
dig example.com AAAA
```

liefert keinen entsprechenden IPv6-Record.

Dann liegt nicht automatisch ein allgemeiner DNS-Ausfall vor.

Die beiden Record-Typen müssen getrennt bewertet werden.

Beispiel: Resolver liefert alte Daten

Ein rekursiver Resolver liefert eine andere Adresse als der autoritative Nameserver.

Dann sollten insbesondere:

TTL,

Cache,

autoritative Daten

und Zeitpunkt einer möglichen DNS-Änderung

untersucht werden.

Beispiel: Delegationsproblem

Ein Name funktioniert über bestimmte Resolver nicht zuverlässig.

Mit:

```
dig +trace example.com
```

kann die Delegationskette untersucht werden.

Dadurch lässt sich erkennen, welche Nameserver auf den einzelnen Ebenen referenziert werden und an welcher Stelle die iterative Auflösung scheitert.

Beispiel: UDP/TCP-Unterschied

Zuerst:

```
dig example.com
```

danach:

```
dig +tcp example.com
```

Wenn nur die TCP-Abfrage funktioniert, sollte der DNS-Transportpfad genauer untersucht werden.

Mit TCPDump oder Wireshark kann parallel geprüft werden, welche DNS-Pakete tatsächlich gesendet und empfangen werden.

Dig und Wireshark

Wireshark kann die von Dig erzeugten DNS-Pakete detailliert analysieren.

Ein Display Filter:

```
dns
```

zeigt den entsprechenden DNS-Verkehr.

Damit kann kontrolliert werden:

welche Anfrage Dig tatsächlich gesendet hat,

an welchen Server,

welcher Record-Typ angefordert wurde,

welche Antwort zurückkam,

und welche DNS-Flags beziehungsweise Records enthalten sind.

Dig und TCPDump

TCPDump ermöglicht die terminalbasierte Beobachtung der DNS-Kommunikation.

Ein typischer Filter lautet:

```
sudo tcpdump -nn port 53
```

Damit können klassische DNS-Abfragen und Antworten über TCP beziehungsweise UDP Port 53 beobachtet werden.

Bei DNS over TLS auf Port 853 ist ein entsprechend anderer Capture-Filter erforderlich.

Dig und MTR

Dig beantwortet die Frage:

Was antwortet DNS?

MTR beantwortet die Frage:

Wie verhält sich der Netzwerkpfad zu einem Ziel?

Antwortet ein DNS-Server nicht, kann MTR ergänzend Hinweise auf den IP-Pfad liefern.

MTR ersetzt jedoch keine DNS-Analyse.

Dig und Nmap

Nmap kann prüfen, ob ein DNS-Dienst auf einem Zielsystem netzwerkseitig sichtbar ist.

Dig kann anschließend eine echte DNS-Abfrage an diesen Dienst richten.

Damit ergänzen sich:

Nmap → Ist ein Dienst erreichbar beziehungsweise erkennbar?

Dig → Welche DNS-Antwort liefert der Nameserver?

Dig und ARP Scan

Bei einem lokalen DNS-Server kann ARP Scan zunächst helfen festzustellen, ob dessen IPv4-System auf dem lokalen Layer-2-Netz sichtbar ist.

Dig untersucht anschließend den DNS-Dienst selbst.

Damit werden unterschiedliche Ebenen getrennt:

ARP → lokale Nachbarschaft

IP/Transport → Erreichbarkeit

DNS → Namensauflösung

Grenzen

Dig prüft DNS.

Es beweist nicht, dass ein Webserver, Mailserver oder anderer Dienst tatsächlich funktioniert.

Ein vorhandener A- oder AAAA-Record beweist nicht die Erreichbarkeit der veröffentlichten Adresse.

Ein MX-Record beweist nicht die Erreichbarkeit des Mailservers.

Ein PTR-Record beweist nicht eine symmetrische Forward-Auflösung.

Eine DNS-Antwort kann aus einem Resolvercache stammen.

Unterschiedliche Resolver können zeitweise unterschiedliche Daten liefern.

`+dnssec` allein ist nicht mit einer vollständigen unabhängigen DNSSEC-Validierung gleichzusetzen.

Ein Exitcode 0 bedeutet ausdrücklich nicht, dass der angefragte Name existiert; NXDOMAIN ist ebenfalls eine empfangene DNS-Antwort.

Sicherheitshinweise

Nur autorisierte DNS-Server gezielt untersuchen.

Zonentransfers nur in Umgebungen durchführen, in denen dies ausdrücklich zulässig ist.

Bei Vergleichen immer dokumentieren, welcher Resolver abgefragt wurde.

A- und AAAA-Records bei Dual-Stack-Problemen getrennt prüfen.

DNS-Erreichbarkeit nicht mit Dienst-Erreichbarkeit gleichsetzen.

`+trace` nicht mit einer normalen rekursiven Resolverabfrage verwechseln.

Bei Skripten NXDOMAIN und andere DNS-Statuswerte zusätzlich zum Prozess-Exitcode auswerten.

DNS-Ausgaben können interne Hostnamen, IP-Adressen und Infrastrukturinformationen enthalten und sollten entsprechend geschützt werden.

Technische Einordnung in JUMBO BLOCK Rescue

Dig übernimmt innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue die Rolle der **gezielten DNS- und Resolveranalyse**.

Die Netzwerkwerkzeuge decken damit inzwischen klar getrennte Ebenen ab:

A.17 Nmap

aktive Host-, Port- und Dienstanalyse.

A.18 Wireshark

grafische Paket- und Protokollanalyse.

A.19 TCPDump

terminalbasierte Paketaufzeichnung und Filterung.

A.20 ARP Scan

lokale IPv4-Geräteerkennung auf Layer 2.

A.21 MTR

fortlaufende Pfad-, Latenz- und Paketverlustanalyse.

A.22 Dig

DNS-, Resolver-, Resource-Record- und Delegationsanalyse.

Ein besonders leistungsfähiger Rescue-Ablauf bei einem Namensauflösungsproblem lautet damit:

ARP Scan → Ist der lokale DNS-Server auf Layer 2 sichtbar?

MTR → Ist sein IP-Pfad erreichbar und auffällig?

Dig → Welche DNS-Antwort liefert er tatsächlich?

TCPDump → Werden Anfrage und Antwort auf dem Netzwerk sichtbar?

Wireshark → Was enthalten die DNS-Pakete im Detail?

Dig schafft damit eine entscheidende Trennung:

Netzwerk funktioniert

bedeutet nicht automatisch:

DNS funktioniert.

Und:

DNS liefert eine Adresse

bedeutet nicht automatisch:

Der adressierte Dienst funktioniert.

Genau diese Trennung der Diagnoseebenen macht Dig zu einem wichtigen Bestandteil von JUMBO BLOCK Rescue.

A.23 Whois

Bereich: Netzwerk
Programm: Whois
Baseline 20261001: Whois 5.6.3
Installiertes Paket: `whois 5.6.3`
Bedienung: Kommandozeile
Aufgabe: Abfrage öffentlicher WHOIS-Verzeichnisdienste für Domains, IP-Netzbereiche, AS-Nummern und weitere Registry-Objekte

Zweck

Whois ist ein Kommandozeilenclient für WHOIS-Verzeichnisdienste.

Die installierte Version sucht nach Objekten in Datenbanken, die über das WHOIS-Protokoll gemäß RFC 3912 erreichbar sind.

WHOIS-Datenbanken können – abhängig von Registry, Objektart und deren Veröffentlichungsregeln – Informationen über:

Domains,

IP-Adressbereiche,

IPv4- und IPv6-Netze,

Autonomous-System-Nummern,

Netzwerkorganisationen,

Registrare,

technische beziehungsweise administrative Registry-Objekte

und Kontakt- oder Abuse-Informationen

bereitstellen.

Welche Informationen tatsächlich zurückgegeben werden, bestimmt der jeweils angesprochene WHOIS-Dienst.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich Whois insbesondere zur:

Zuordnung öffentlicher IP-Adressen zu registrierten Netzbereichen,

Ermittlung der für einen IP-Block zuständigen Registry beziehungsweise Organisation,

Untersuchung von Autonomous-System-Nummern,

Abfrage von Domain-Registrierungsinformationen,

Ermittlung zuständiger Registrar- oder Registry-Systeme,

Suche nach veröffentlichten Abuse-Kontakten,

Einordnung unbekannter öffentlicher Netzwerkziele,

und Ergänzung von Nmap-, MTR-, Dig-, TCPDump- und Wireshark-Ergebnissen um Registryinformationen.

WHOIS ist kein DNS

WHOIS und DNS erfüllen unterschiedliche Aufgaben.

DNS beantwortet beispielsweise:

Welche IP-Adresse gehört zu einem Namen?

WHOIS kann dagegen beispielsweise Hinweise darauf liefern:

Welchem registrierten Netzbereich ist diese öffentliche IP-Adresse zugeordnet?

Dig und Whois ergänzen sich deshalb, ersetzen sich aber nicht.

WHOIS ist keine Geo-IP-Datenbank

WHOIS-Daten können Adress- oder Länderangaben enthalten.

Daraus darf jedoch nicht automatisch der tatsächliche physische Standort eines Servers oder Endgeräts abgeleitet werden.

Eine registrierte Organisation kann ihre Infrastruktur an anderen Standorten betreiben.

Netze können:

international genutzt,

weiterdelegiert,

an Kunden vergeben,

über Cloud-Infrastrukturen verwendet

oder an anderen Orten geroutet

werden.

WHOIS ist keine Personensuche

WHOIS ist technisch ein Registry- beziehungsweise Verzeichnisabfragesystem.

Es sollte nicht als Werkzeug verstanden werden, mit dem sich zuverlässig die natürliche Person hinter:

einer Domain,

einer IP-Adresse

oder einem Server

bestimmen lässt.

Insbesondere moderne Domainregistrierungen enthalten häufig datenschutzbedingt redigierte oder durch Proxy-/Privacy-Dienste ersetzte Kontaktinformationen.

Grundsyntax

Der grundlegende Aufruf lautet:

```
whois OBJECT
```

Das Objekt kann je nach Abfrage beispielsweise sein:

Domainname,

IPv4-Adresse,

IPv6-Adresse,

AS-Nummer,

NIC-Handle

oder ein anderer vom betreffenden WHOIS-Dienst unterstützter Objekttyp.

Automatische Serverauswahl

Whois 5.6.3 versucht selbst zu bestimmen, welcher WHOIS-Server für das angegebene Objekt geeignet ist.

Dadurch genügt für viele Standardabfragen:

```
whois OBJECT
```

ohne dass der Benutzer den Registryserver vorher kennen muss.

Domain abfragen

Beispiel:

```
whois example.com
```

Der Client versucht, den für diese Domain geeigneten WHOIS-Dienst zu bestimmen und die Anfrage dorthin zu richten.

Welche Daten zurückgegeben werden, hängt von:

Top-Level-Domain,

Registry,

Registrar,

WHOIS-Modell,

Datenschutzregeln

und Serverimplementierung

ab.

IP-Adresse abfragen

Auch eine öffentliche IPv4-Adresse kann unmittelbar abgefragt werden:

```
whois 192.0.2.1
```

Die hier verwendete Adresse dient ausschließlich als Dokumentationsbeispiel.

Bei einer realen öffentlichen Adresse kann die Antwort beispielsweise Informationen zum registrierten Netzbereich und zur zuständigen Organisation enthalten.

IPv6-Adresse

Whois kann ebenfalls IPv6-Objekte abfragen.

Beispielprinzip:

```
whois IPv6-ADRESSE
```

Der Client bestimmt dabei nach seinen internen Regeln den geeigneten WHOIS-Dienst.

AS-Nummer

Autonomous-System-Nummern können beispielsweise in der Form:

```
AS64496
```

abgefragt werden.

Eine WHOIS-Antwort kann abhängig von der Registry Informationen zum betreffenden AS-Objekt enthalten.

Autonomous System

Ein Autonomous System ist im Internet-Routing ein administrativ kontrollierter Routingbereich, dem eine Autonomous-System-Nummer – ASN – zugeordnet ist.

WHOIS-Daten können helfen, eine in Netzwerkdiagnosen auftretende ASN einer registrierten Organisation beziehungsweise einem Registryobjekt zuzuordnen.

MTR und Whois

MTR kann den Netzwerkpfad zu einem entfernten Ziel sichtbar machen.

Whois kann anschließend zur Einordnung öffentlicher IP-Adressen beziehungsweise AS-Informationen einzelner Netzbereiche verwendet werden.

Damit entsteht die Kombination:

MTR → Wo verläuft der sichtbare IP-Pfad?

Whois → Welchen registrierten Netzobjekten sind öffentliche Adressen beziehungsweise AS-Objekte zugeordnet?

Bestimmten WHOIS-Server auswählen

Mit:

`-h HOST`

beziehungsweise:

`--host=HOST`

wird ein bestimmter WHOIS-Server ausdrücklich ausgewählt.

Grundprinzip:

`whois -h SERVER OBJECT`

Damit kann die automatische Serverauswahl umgangen werden.

Warum einen Server ausdrücklich auswählen?

Dies ist nützlich, wenn:

eine bestimmte Registry direkt abgefragt werden soll,

mehrere WHOIS-Quellen verglichen werden sollen,

ein Referral überprüft werden soll,

oder bekannt ist, dass ein bestimmtes Objekt in einer bestimmten Datenbank untersucht werden muss.

Port festlegen

Mit:

`-p PORT`

beziehungsweise:

`--port=PORT`

kann der Zielport ausdrücklich angegeben werden.

Klassisches WHOIS nach RFC 3912 verwendet TCP Port:

43.

Ein abweichender Port kann für besondere beziehungsweise nichtstandardisierte Dienste angegeben werden.

WHOIS verwendet TCP

Klassisches WHOIS arbeitet über eine TCP-Verbindung.

Der Client verbindet sich mit dem WHOIS-Server, sendet die Anfrage und empfängt die textuelle Antwort.

Das unterscheidet WHOIS beispielsweise von klassischen DNS-Standardabfragen, die häufig zunächst UDP verwenden.

IANA zuerst abfragen

Whois 5.6.3 besitzt die Option:

`-I`

Damit wird zunächst:

`whois.iana.org`

abgefragt.

Anschließend folgt der Client dem von IANA gelieferten Referral zum für die Anfrage autoritativen WHOIS-Server.

Diese Funktion unterstützt:

IP-Adressen,

AS-Nummern

und Domains.

Datenschutz bei `-I`

Die Dokumentation weist ausdrücklich darauf hin:

Bei Verwendung von `-I` erhält der IANA-WHOIS-Server die vollständige Anfrage.

Dies sollte berücksichtigt werden, wenn Abfragen sensible interne Untersuchungszusammenhänge erkennen lassen könnten.

Referral

WHOIS ist historisch auf verschiedene Registries und Registrare verteilt.

Ein Server kann deshalb auf einen anderen WHOIS-Dienst verweisen.

Ein solcher Verweis wird als Referral bezeichnet.

Der Client kann entsprechenden Referral-Informationen folgen.

Registry und Registrar

Bei Domainabfragen muss zwischen Registry und Registrar unterschieden werden.

Die **Registry** verwaltet die zentrale Registrierungsdatenbank einer Top-Level-Domain beziehungsweise des entsprechenden Namensraums.

Der **Registrar** ist der Anbieter, über den eine konkrete Domain registriert beziehungsweise verwaltet wird.

Je nach WHOIS-Modell können Registry- und Registrar-Daten auf unterschiedlichen Servern liegen.

Thin Registry

Bei bestimmten gTLD-WHOIS-Systemen können Registryserver nur einen Teil der Registrierungsinformationen liefern und auf den zuständigen Registrar verweisen.

Whois 5.6.3 besitzt dafür spezielle Referral-Logik.

Referral-Rekursion deaktivieren

Mit:

```
--no-recursion
```

kann die Rekursion von Registry- zu Registrarservern deaktiviert werden.

Dies ist hilfreich, wenn ausdrücklich nur die Antwort der zunächst angesprochenen Registry betrachtet werden soll.

Warum Referral-Kontrolle wichtig ist

Ohne diese Unterscheidung kann eine WHOIS-Ausgabe so wirken, als stamme sie vollständig aus einer einzigen Datenquelle.

Tatsächlich kann der Client jedoch einem Referral gefolgt sein.

Für eine dokumentierte Untersuchung sollte deshalb berücksichtigt werden, welche Quelle die jeweilige Information geliefert hat.

Ausführliche Programmausgabe

Mit:

```
--verbose
```

arbeitet der Client ausführlicher.

Dies kann bei der technischen Diagnose der WHOIS-Abfrage hilfreich sein.

Rechtliche Hinweise ausblenden

Mit:

```
-H
```

werden bestimmte rechtliche Hinweise beziehungsweise Disclaimer unterdrückt, die einige Registries in ihren Antworten ausgeben.

Das verändert nicht die Registrydaten selbst, sondern deren Darstellung durch den Client.

Registry-Hinweise können wichtig sein

Bei dokumentations- oder compliancebezogenen Untersuchungen sollte nicht automatisch jeder Disclaimer entfernt werden.

Registryhinweise können Informationen enthalten über:

Nutzungsbedingungen,

Weiterverwendung,

Abfragelimits

oder Interpretation der Daten.

RIPE-artige WHOIS-Dienste

Whois 5.6.3 unterstützt zusätzlich eine Reihe von Optionen, die von:

`whois.ripe.net`

und einigen ähnlich arbeitenden WHOIS-Diensten verstanden werden.

Diese Optionen sind **nicht universell auf alle WHOIS-Server übertragbar**.

Wichtige Konsequenz

Eine Option, die bei RIPE funktioniert, muss bei einer anderen Registry nicht dieselbe Bedeutung besitzen oder überhaupt unterstützt werden.

WHOIS ist hinsichtlich der Abfragesprache nicht weltweit vollständig vereinheitlicht.

Kurze IP-/Abuse-Ausgabe

Bei RIPE-artigen Servern kann:

`-b`

eine kurze Darstellung von IP-Adressbereichen mit Abuse-Kontakt anfordern.

Dies kann für eine schnelle Einordnung eines öffentlichen Netzbereichs nützlich sein.

Die Option ist jedoch serverabhängig.

Objektfilterung deaktivieren

Bei unterstützenden RIPE-artigen Servern kann:

`-B`

die serverseitige Objektfilterung deaktivieren.

Die Manpage beschreibt dies unter anderem als Möglichkeit, E-Mail-Adressen anzuzeigen.

Ob und welche Daten tatsächlich zurückgegeben werden, bestimmt der Server.

Die Option hebt keine serverseitigen Datenschutz- oder Zugriffsregeln auf.

Exakte Netzpräfixsuche

Mit:

`-x`

kann bei unterstützenden RIPE-artigen WHOIS-Diensten nach einer exakten Übereinstimmung eines Netzwerkpräfixes gesucht werden.

Dies ist insbesondere bei verschachtelten Netzzuweisungen nützlich.

Weniger spezifische Objekte

Für unterstützende Server existieren Optionen zur Suche nach weniger spezifischen Netzobjekten.

`-l`

liefert eine Ebene weniger spezifisch.

`-L`

liefert alle Ebenen weniger spezifischer Objekte.

Damit kann die Hierarchie einer Netzzuweisung untersucht werden.

Spezifischere Objekte

Entsprechend stehen:

`-m`

für eine Ebene spezifischere Objekte

und:

`-M`

für alle Ebenen spezifischerer Objekte

zur Verfügung.

Dies kann bei delegierten beziehungsweise unterteilten Netzbereichen hilfreich sein.

Warum Netz-Hierarchien relevant sind

Eine öffentliche IP-Adresse kann Teil eines großen Providerblocks sein, innerhalb dessen ein kleinerer Bereich einem Kunden oder Dienst zugeordnet wurde.

Nur die Betrachtung des größten Netzblocks kann deshalb zu einer zu groben Zuordnung führen.

Weniger und stärker spezifische Objekte helfen, diese Hierarchie zu untersuchen.

Quellen auswählen

Mit:

`-s SOURCE[,SOURCE]...`

kann bei RIPE-artigen Servern festgelegt werden, welche gespiegelten Datenquellen durchsucht werden sollen.

Die Reihenfolge der Quellen ist relevant.

Welche Quellen verfügbar sind, kann über eine Serverabfrage ermittelt werden.

Serverinformationen

Mit:

`-q KEYWORD`

können unterstützende RIPE-artige Server nach Informationen über sich selbst gefragt werden.

Dokumentierte Schlüsselwörter sind unter anderem:

`version`

`sources`

`types`.

Objekttypen

Mit:

`-T TYPE[,TYPE]...`

kann die Suche bei unterstützenden Servern auf bestimmte Objekttypen eingeschränkt werden.

Dies kann große oder mehrdeutige Ergebnisse präzisieren.

Objektvorlagen

Mit:

`-t TYPE`

kann die Vorlage eines Objekttyps abgefragt werden.

Mit:

`-v TYPE`

kann eine ausführlichere Vorlage angefordert werden.

Diese Funktionen beziehen sich auf Server, die die entsprechende RIPE-artige Abfragesprache unterstützen.

Inverse Suche

Mit:

```
-i ATTR[,ATTR]...
```

können bei unterstützenden Servern Objekte gesucht werden, deren angegebene Attribute einen bestimmten Wert besitzen.

Dies ist eine weiterführende Registrydatenbankfunktion und keine allgemeine Suchfunktion über das Internet.

Kontakt-Rekursion deaktivieren

Bei RIPE-artigen Abfragen kann:

```
-r
```

rekursive Suchvorgänge nach Kontaktinformationen deaktivieren.

Dies unterscheidet sich von:

```
--no-recursion
```

welches die Rekursion von Registry- zu Registrarservern betrifft.

Die beiden Optionen dürfen daher nicht verwechselt werden.

Referral auf RIPE-artigen Servern

Mit:

```
-R
```

kann bei entsprechenden Servern das Folgen bestimmter Referrals deaktiviert und stattdessen die lokale Kopie des Objekts angezeigt werden.

Auch diese Option ist serverspezifisch.

Konfigurationsdatei

Whois 5.6.3 berücksichtigt:

```
/etc/whois.conf
```

sofern diese Datei vorhanden ist.

Damit können alternative WHOIS-Server definiert werden, die die eingebauten Regeln ergänzen oder überschreiben.

Aufbau von `whois.conf`

Die Datei ordnet Suchmuster bestimmten WHOIS-Servern zu.

Vereinfacht:

```
MUSTER WHOIS-SERVER
```

Das Muster wird gegen das abzufragende WHOIS-Objekt geprüft.

Bei einem Treffer wird der zugeordnete Server verwendet.

Bedeutung für Rescue-Diagnosen

Eine lokale `whois.conf` kann beeinflussen, welcher WHOIS-Server tatsächlich abgefragt wird.

Bei unerwarteten Ergebnissen sollte daher berücksichtigt werden, dass die automatische Serverwahl lokal überschrieben sein kann.

Aus der Paketbaseline allein wird nicht abgeleitet, ob JUMBO BLOCK Rescue eine angepasste `/etc/whois.conf` besitzt.

Umgebungsvariable `WHOIS_SERVER`

Die Variable:

`WHOIS_SERVER`

kann einen Server vorgeben, wenn Whois selbst nicht bestimmen kann, wo das betreffende Objekt zu finden ist.

Ist diese Variable nicht gesetzt und kann der Client keinen passenden Server bestimmen, sieht die dokumentierte Fallback-Logik für bestimmte Objekte weitere Standardserver vor.

Umgebungsvariable `WHOIS_OPTIONS`

Mit:

`WHOIS_OPTIONS`

können Optionen definiert werden, die vor den explizit auf der Kommandozeile angegebenen Optionen ausgewertet werden.

Dies kann das Verhalten eines Aufrufs verändern.

Für reproduzierbare Untersuchungen sollten solche Umgebungseinflüsse berücksichtigt werden.

Internationalisierte Domains

Whois 5.6.3 unterstützt die Verarbeitung internationalisierter Domainnamen.

Kommandozeilenargumente werden entsprechend der aktuellen Locale interpretiert und für IDN-Domains in eine ASCII-kompatible Darstellung umgesetzt.

ACE / Punycode

WHOIS-Konfigurationen verwenden für internationalisierte Domainnamen die ASCII Compatible Encoding – ACE.

Solche Namen sind häufig an der:

`xn--`

Darstellung erkennbar.

Zeichenkodierung

Das klassische WHOIS-Protokoll definiert keine einheitliche Zeichenkodierung für Zeichen außerhalb von ASCII.

Die Manpage weist deshalb darauf hin, dass Implementierungen in diesem Bereich stark variieren.

Whois versucht bei bekannten Servern gegebenenfalls, die Ausgabe passend zur aktuellen System-Locale umzuwandeln.

WHOIS-Daten sind nicht einheitlich

Anders als beispielsweise DNS Resource Records besitzen WHOIS-Antworten weltweit kein einheitliches, überall identisches Ausgabeformat.

Unterschiedliche Registries können:

andere Feldnamen,

andere Objektmodelle,

andere Referralmechanismen,

andere Datenschutzregeln

und andere Textformate

verwenden.

Automatische Auswertung ist daher schwierig

Ein Shell-Skript sollte WHOIS-Ausgaben nicht ohne Weiteres so behandeln, als hätten alle Registries dieselbe Struktur.

Ein Parser, der für eine Registry funktioniert, kann bei einer anderen fehlschlagen.

Domain-WHOIS und Datenschutz

Bei Domainregistrierungen können personenbezogene Felder:

redigiert,

durch Privacy-Dienste ersetzt,

nicht veröffentlicht

oder nur über andere geregelte Verfahren verfügbar

sein.

Das Fehlen eines Namens oder einer Adresse ist daher nicht automatisch ein technischer Fehler.

Registrant ist nicht zwingend Betreiber

Selbst wenn ein WHOIS-Dienst Registrierungsinformationen anzeigt, darf daraus nicht automatisch geschlossen werden, dass die genannte Organisation:

den Server physisch betreibt,

den Inhalt einer Website erstellt,

den Netzwerkverkehr verursacht,

oder die betreffende Infrastruktur technisch administriert.

Domaininhaber, Registrar, Hostinganbieter, CDN, Cloudanbieter und tatsächlicher Dienstbetreiber können unterschiedliche Stellen sein.

IP-WHOIS und Netzzuordnung

Bei öffentlichen IP-Adressen beschreibt WHOIS typischerweise die Registrierung beziehungsweise Zuweisung eines Netzbereichs.

Dies ist besonders hilfreich, um festzustellen, ob eine Adresse beispielsweise einem:

Internetprovider,

Hostinganbieter,

Cloudnetz,

Unternehmen

oder anderen registrierten Netzbetreiber

zugeordnet ist.

Die genaue Aussagekraft hängt vom jeweiligen Registryobjekt ab.

IP-Zuordnung beweist keine Benutzeridentität

Aus der registrierten Organisation eines IP-Netzblocks lässt sich nicht automatisch bestimmen, welcher konkrete Benutzer eine einzelne IP-Adresse zu einem bestimmten Zeitpunkt verwendet hat.

Das gilt insbesondere bei:

dynamischen IP-Adressen,

NAT,

Carrier-Grade NAT,

VPN,

Proxy,

Cloudsystemen

und gemeinsam genutzter Infrastruktur.

ASN-Zuordnung

WHOIS kann Informationen zu registrierten Autonomous-System-Objekten liefern.

Diese Angaben helfen bei der administrativen Einordnung eines AS.

Sie ersetzen jedoch keine Echtzeit-BGP-Analyse.

Registrierung und tatsächliches Routing

Ein Registryeintrag und die aktuell im Internet sichtbare Routingrealität sind unterschiedliche Informationsquellen.

Ein Netz kann beispielsweise über:

andere Upstream-Provider,

mehrere Autonomous Systems,

Anycast

oder komplexe Routingbeziehungen

erreichbar sein.

WHOIS beschreibt Registryinformationen, nicht automatisch den vollständigen aktuellen Routingzustand.

Whois und Dig

Eine sinnvolle Kombination lautet:

```
dig example.com A
```

ermittelt die veröffentlichte IPv4-Adresse.

Anschließend:

```
whois IP-ADRESSE
```

liefert Registryinformationen zum betreffenden öffentlichen Netzbereich.

Damit wird getrennt:

DNS → Name zu Adresse

WHOIS → Adresse zu registriertem Netzobjekt

Whois und MTR

MTR kann öffentliche Routeradressen entlang eines Pfads sichtbar machen.

Whois kann einzelne dieser Adressen beziehungsweise zugehörige Netzobjekte administrativ einordnen.

Dabei muss berücksichtigt werden:

Ein MTR-Hop kann eine Interface-Adresse anzeigen, deren WHOIS-Zuordnung nicht unmittelbar den tatsächlichen Betreiber des gesamten Pfadsegments beschreibt.

Whois und Nmap

Nmap kann einen öffentlichen Host technisch untersuchen.

Whois kann die öffentliche IP-Adresse registryseitig einordnen.

Vereinfacht:

Nmap → Was ist netzwerktechnisch am Ziel sichtbar?

Whois → Welcher Registryzuordnung gehört der öffentliche Adressbereich an?

Whois und Wireshark/TCPDump

Wireshark oder TCPDump können eine unbekannte öffentliche Gegenstelle im Netzwerkverkehr sichtbar machen.

Whois kann anschließend zur administrativen Einordnung der betreffenden öffentlichen IP-Adresse verwendet werden.

Ein sinnvoller Analyseablauf lautet:

Capture → unbekannte IP feststellen → Whois-Abfrage → Registryzuordnung dokumentieren

Whois und ARP Scan

ARP Scan arbeitet im unmittelbar lokalen IPv4-Layer-2-Netz und zeigt MAC-Adressen.

Whois arbeitet mit öffentlichen Registrydaten.

Private lokale Adressen wie RFC-1918-Adressen besitzen keine individuelle öffentliche WHOIS-Zuordnung zu einem lokalen Endgerät.

Für ein lokales Gerät sind daher beispielsweise:

ARP,

DHCP,

Switchinformationen

oder lokale Inventardaten

relevanter als eine öffentliche WHOIS-Abfrage.

Typischer Rescue-Ablauf: unbekannte öffentliche IP-Adresse

1. IP-Adresse dokumentieren

Beispielsweise aus:

TCPDump,

Wireshark,

Nmap

oder einem Log.

2. WHOIS-Abfrage durchführen

```
whois IP-ADRESSE
```

3. Netzbereich identifizieren

Prüfen, welchem registrierten Präfix beziehungsweise Netzobjekt die Adresse zugeordnet ist.

4. Organisation und Registry prüfen

5. Falls vorhanden Referral beziehungsweise spezifischeres Objekt untersuchen

6. Ergebnisse nicht mit einer persönlichen Identität gleichsetzen

7. Bei Bedarf technische Analyse mit Nmap, MTR oder Dig fortsetzen

Typischer Rescue-Ablauf: unbekannte Domain

1. DNS prüfen

```
dig DOMAIN
```

2. Domain-WHOIS prüfen

```
whois DOMAIN
```

3. Registrar beziehungsweise Registryinformationen dokumentieren

4. Aufgelöste öffentliche IP-Adresse separat untersuchen

```
whois IP-ADRESSE
```

5. Domainregistrierung und IP-Netzzuordnung getrennt bewerten

Eine Domain kann bei Organisation A registriert sein und auf Infrastruktur von Organisation B zeigen.

Typischer Rescue-Ablauf: MTR-Hop einordnen

MTR zeigt eine unbekannte öffentliche Routeradresse.

Diese Adresse kann mit:

```
whois IP-ADRESSE
```

abgefragt werden.

Dadurch kann ein Hinweis auf den registrierten Netzbetreiber entstehen.

Bei mehreren aufeinanderfolgenden Hops können dadurch administrative Netzgrenzen sichtbarer werden.

Die WHOIS-Zuordnung ersetzt jedoch keine vollständige Routinganalyse.

Typischer Rescue-Ablauf: AS-Nummer

Ist eine AS-Nummer bekannt:

```
whois ASNUMMER
```

kann das entsprechende Registryobjekt abgefragt werden.

Dies kann Informationen über die registrierte Organisation und weitere technische Registrydaten liefern.

Private IPv4-Adressen

Private IPv4-Adressbereiche sind nicht wie öffentliche Internetadressen einzelnen lokalen Organisationen über das globale WHOIS-System zugeordnet.

Eine Abfrage einer privaten Adresse kann daher höchstens Informationen über den reservierten Adressbereich liefern, nicht darüber, welches lokale Gerät im untersuchten Netzwerk diese Adresse verwendet.

Loopback und reservierte Adressen

Dasselbe Grundprinzip gilt für andere reservierte oder spezielle Adressbereiche.

WHOIS ist keine lokale Inventardatenbank.

Offline-Betrieb

Whois benötigt für eine aktuelle externe WHOIS-Abfrage eine Netzwerkverbindung zum betreffenden WHOIS-Server.

Ohne Netzwerkzugang kann der Client keine aktuelle Antwort einer externen Registry abrufen.

Damit unterscheidet er sich beispielsweise von Werkzeugen, die ausschließlich lokale Daten untersuchen.

WHOIS-Server kann nicht erreichbar sein

Eine fehlgeschlagene WHOIS-Abfrage beweist nicht:

dass das Objekt nicht existiert.

Mögliche Ursachen sind beispielsweise:

keine Netzwerkverbindung,

WHOIS-Port blockiert,

Server nicht erreichbar,

Rate Limiting,

falscher Server,

oder serverseitige Einschränkungen.

Rate Limits

WHOIS-Dienste können Abfragen begrenzen.

Zu viele automatisierte Anfragen können:

gedrosselt,

temporär blockiert

oder abgewiesen

werden.

Whois sollte deshalb nicht für unkontrollierte Massensuchen verwendet werden.

Datenschutz und Nutzungsbedingungen

WHOIS-Antworten können personenbezogene oder kontaktbezogene Informationen enthalten.

Die Tatsache, dass Daten öffentlich abrufbar sind, bedeutet nicht automatisch, dass sie ohne Einschränkung für jeden Zweck weiterverarbeitet oder veröffentlicht werden dürfen.

Registryhinweise und geltende Datenschutz- beziehungsweise Nutzungsregeln sind zu beachten.

Daten können veraltet sein

WHOIS-Daten sind Registrydaten.

Sie können:

veraltet,

unvollständig,

redigiert

oder aufgrund von Delegationen nur indirekt aussagekräftig

sein.

Ein WHOIS-Ergebnis sollte daher als Informationsquelle dokumentiert, aber nicht überinterpretiert werden.

WHOIS und RDAP

Viele Registries stellen inzwischen zusätzlich beziehungsweise zunehmend RDAP – Registration Data Access Protocol – bereit.

Der hier dokumentierte Befehl `whois` ist jedoch ein Client für den klassischen WHOIS-Verzeichnisdienst.

Die Existenz moderner RDAP-Dienste ändert nicht die Funktion des in der Baseline installierten Whois-Clients.

Grenzen

Whois liefert Registry- beziehungsweise Verzeichnisdaten.

Es führt keinen Portscan durch.

Es analysiert keine Netzwerkpakete.

Es löst keine DNS-Probleme.

Es bestimmt nicht zuverlässig den physischen Standort eines Servers.

Es identifiziert nicht zuverlässig den tatsächlichen Benutzer einer IP-Adresse.

Es beweist nicht den Betreiber einer Website.

Es beweist nicht das Eigentum an einem physischen System.

Es liefert keine vollständige Echtzeitansicht des Internet-Routings.

WHOIS-Ausgabeformate und verfügbare Felder unterscheiden sich zwischen Registries.

Datenschutzregeln können wesentliche Felder ausblenden.

Sicherheitshinweise

WHOIS-Ergebnisse immer als Registryinformationen und nicht als Identitätsbeweis behandeln.

Registrierte Organisation nicht automatisch mit tatsächlichem Benutzer oder technischem Betreiber gleichsetzen.

Domainregistrierung und Hostinginfrastruktur getrennt untersuchen.

Bei IP-Adressen Netzblockzuordnung und konkrete Endgeräteidentität nicht verwechseln.

Bei MTR-Hops WHOIS-Zuordnung nicht als vollständigen Beweis der Routingverantwortung interpretieren.

Referral-Quellen bei wichtigen Untersuchungen dokumentieren.

WHOIS-Antworten können personenbezogene Daten enthalten.

Rate Limits und Nutzungsbedingungen der Registry beachten.

Bei automatisierter Verarbeitung unterschiedliche Serverformate berücksichtigen.

Technische Einordnung in JUMBO BLOCK Rescue

Whois übernimmt innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue die Rolle der **administrativen und registrybasierten Einordnung öffentlicher Netzwerkobjekte**.

Damit ergibt sich inzwischen eine sehr klare Arbeitsteilung:

A.17 Nmap

aktive Host-, Port- und Dienstanalyse.

A.18 Wireshark

grafische Paket- und Protokollanalyse.

A.19 TCPDump

terminalbasierte Paketaufzeichnung und Filterung.

A.20 ARP Scan

lokale IPv4-Geräteerkennung auf Layer 2.

A.21 MTR

fortlaufende Pfad-, Latenz- und Paketverlustanalyse.

A.22 Dig

DNS-, Resolver- und Delegationsanalyse.

A.23 Whois

Registry-, Netzblock-, Domain- und AS-Einordnung.

Ein typischer kombinierter Rescue-Workflow kann damit lauten:

Dig → Welche Adresse liefert DNS?

Whois → Welchem öffentlichen Registry-Netzbereich ist diese Adresse zugeordnet?

MTR → Über welchen sichtbaren Netzwerkpfad wird das Ziel erreicht?

Nmap → Welche Ports und Dienste sind am autorisierten Ziel sichtbar?

TCPDump/Wireshark → Welche Kommunikation findet tatsächlich statt?

Whois liefert dabei bewusst eine andere Informationsklasse als die übrigen Werkzeuge:

Es untersucht nicht primär das Verhalten des Netzwerks.

Es fragt die **registrierten Verwaltungs- und Zuordnungsdaten** zu Netzwerkobjekten ab.

Für die korrekte Interpretation gilt deshalb als zentrale Regel:

WHOIS sagt, was eine Registry über ein Objekt veröffentlicht – nicht automatisch, wer ein konkretes System tatsächlich betreibt, benutzt oder kontrolliert.

A.24 Chromium

Bereich: Netzwerk
Programm: Chromium
Baseline 20261001: Chromium 154.0.8037.57
Installierte Pakete: `chromium 154.0.8037.57-1~deb13u1`, `chromium-common 154.0.8037.57-1~deb13u1`, `chromium-l10n 154.0.8037.57-1~deb13u1`, `chromium-sandbox 154.0.8037.57-1~deb13u1`
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Webzugriff, Zugriff auf lokale webbasierte Geräteoberflächen sowie HTTP-/HTTPS- und Webanwendungsdiagnose

Zweck

Chromium ist ein Webbrowser und zugleich die Open-Source-Basis der Chromium-Browserfamilie.

Innerhalb von JUMBO BLOCK Rescue besitzt Chromium eine weitergehende diagnostische Funktion.

Viele moderne Netzwerkgeräte und Infrastruktursysteme werden über webbasierte Administrationsoberflächen bedient.

Dazu gehören beispielsweise:

Router,

Firewalls,

Managed Switches,

Access Points,

NAS-Systeme,

Servermanagementsysteme,

Drucker,

USV-Systeme,

Storage-Systeme,

Virtualisierungsplattformen

und andere Appliances.

Chromium ermöglicht den direkten Zugriff auf solche HTTP- beziehungsweise HTTPS-basierten Verwaltungsoberflächen.

Darüber hinaus können die integrierten Developer Tools verwendet werden, um den Webverkehr einer Anwendung aus Sicht des Browsers zu untersuchen.

Einsatzgebiet im Rescue-System

Chromium eignet sich innerhalb von JUMBO BLOCK Rescue insbesondere für:

Zugriff auf öffentliche Webseiten,

Zugriff auf lokale Router- und Appliance-Oberflächen,

HTTP-/HTTPS-Funktionstests,

Prüfung webbasierter Administrationsoberflächen,

Untersuchung von Zertifikatsfehlern,

Analyse von HTTP-Statuscodes,

Analyse von Webrequests,

Kontrolle von Request- und Response-Headern,

Untersuchung von Cookies,

Analyse von Weiterleitungen,

Untersuchung von Ladeproblemen,

Abruf technischer Dokumentation,

Download benötigter Dateien

und ergänzende Diagnose zusammen mit Dig, MTR, Nmap, TCPDump und Wireshark.

Browser und Netzwerkdiagnose

Ein Browser testet mehr als die reine IP-Erreichbarkeit.

Beim Aufruf einer HTTPS-Webseite können unter anderem folgende Komponenten beteiligt sein:

DNS,

IP-Routing,

TCP beziehungsweise andere unterstützte Webtransporte,

TLS,

Zertifikatsprüfung,

HTTP,

Redirects,

Cookies,

JavaScript,

Web APIs

und die eigentliche Webanwendung.

Dadurch kann Chromium Fehler sichtbar machen, die mit einem einfachen Ping oder Portscan nicht erkennbar wären.

Typische Diagnosekette

Der Aufruf:

`https://example.com`

kann vereinfacht mehrere Schritte erfordern:

1. Name auflösen

DNS ermittelt eine Adresse.

2. Ziel erreichen

Das Netzwerk transportiert die Verbindung zum Ziel.

3. Sichere Verbindung aufbauen

Bei HTTPS wird eine verschlüsselte Verbindung aufgebaut und die Identität des Servers anhand der Zertifikatsmechanismen geprüft.

4. HTTP-Kommunikation

Browser und Server tauschen HTTP-Anfragen und Antworten aus.

5. Ressourcen laden

HTML kann weitere Ressourcen anfordern, beispielsweise:

CSS,

JavaScript,

Bilder,

Fonts,

API-Daten.

6. Anwendung ausführen

JavaScript und andere Browserfunktionen erzeugen die eigentliche Webanwendung.

Ein Fehler auf einer dieser Ebenen kann dazu führen, dass die Seite nicht oder nicht vollständig funktioniert.

Direkte IP-Adresse

Lokale Geräte können häufig direkt über ihre IP-Adresse angesprochen werden.

Beispielprinzip:

`http://192.0.2.1`

oder:

`https://192.0.2.1`

Die hier verwendete Adresse dient ausschließlich als Dokumentationsbeispiel.

Bei einer direkten IP-Verbindung wird die normale DNS-Auflösung des Zielnamens umgangen.

Bedeutung für die Diagnose

Funktioniert:

`https://IP-ADRESSE`

aber:

`https://hostname.example`

nicht,

ist dies ein wichtiger Hinweis.

Mögliche Untersuchungsrichtungen sind dann beispielsweise:

DNS,

Namensauflösung,

Proxykonfiguration,

Host-basierte Serverkonfiguration

oder TLS-Zertifikatszuordnung.

Die direkte IP-Verbindung beweist jedoch nicht automatisch, dass alle Funktionen der Webanwendung korrekt arbeiten.

Hostname kann technisch erforderlich sein

Moderne Webserver können mehrere virtuelle Hosts auf derselben IP-Adresse betreiben.

Außerdem können TLS-Zertifikate für bestimmte DNS-Namen ausgestellt sein.

Eine Webseite, die über ihren korrekten Hostnamen funktioniert, muss daher nicht zwingend über die reine IP-Adresse identisch funktionieren.

HTTP

HTTP überträgt Webinhalte ohne die durch HTTPS bereitgestellte Transportverschlüsselung und Serverauthentifizierung.

Bei unverschlüsselten Verbindungen können Daten auf dem Netzwerkpfad grundsätzlich beobachtet oder verändert werden.

Anmeldedaten und andere sensible Informationen sollten deshalb nicht unbedacht über eine ungesicherte HTTP-Verbindung übertragen werden.

HTTPS

HTTPS verbindet HTTP mit einer verschlüsselten Transportverbindung.

Bei einer ordnungsgemäß authentifizierten HTTPS-Verbindung werden die übertragenen Daten zwischen Browser und Gegenstelle gegen einfaches Mitlesen und Verändern auf dem Transportweg geschützt.

HTTPS allein beweist jedoch nicht, dass der Inhalt einer Webseite vertrauenswürdig ist.

Auch eine schädliche Webseite kann grundsätzlich HTTPS verwenden.

Zertifikate

Für die Authentifizierung einer HTTPS-Gegenstelle verwendet der Browser digitale Zertifikate.

Bei der Prüfung spielen unter anderem eine Rolle:

Zertifikatskette,

vertrauenswürdige Zertifizierungsstelle,

Gültigkeitszeitraum,

angeforderter Hostname

und weitere Sicherheitsanforderungen.

Zertifikatsfehler

Chromium kann bei Problemen mit einer sicheren Verbindung Warnungen anzeigen.

Typische Fehlerklassen umfassen beispielsweise:

nicht vertrauenswürdige Zertifizierungsstelle,

Hostname passt nicht zum Zertifikat,

ungültiges beziehungsweise abgelaufenes Zertifikat,

oder andere TLS-/Zertifikatsprobleme.

Solche Warnungen dürfen in einer Rescue-Situation nicht reflexartig ignoriert werden.

Lokale Appliances und selbstsignierte Zertifikate

Lokale Netzwerkgeräte verwenden teilweise:

selbstsignierte Zertifikate,

interne Zertifizierungsstellen,

veraltete Zertifikate

oder Zertifikate, deren Name nicht zur verwendeten IP-Adresse passt.

Eine Warnung kann deshalb bei einer lokalen Appliance einen technisch erklärbaren Grund besitzen.

Sie bleibt dennoch ein Sicherheitsindikator.

Vor dem Umgehen einer Zertifikatswarnung

Vor einem bewussten Fortfahren sollte geprüft werden:

Ist dies wirklich das erwartete Gerät?

Stimmt die IP-Adresse?

Stimmt der Hostname?

Ist das Gerät über einen vertrauenswürdigen lokalen Pfad erreichbar?

Ist die Zertifikatsabweichung für dieses Gerät bekannt und dokumentiert?

Hat sich das Zertifikat unerwartet geändert?

Könnte eine falsche Gegenstelle oder ein Man-in-the-Middle-Szenario vorliegen?

Systemzeit

Die Systemzeit ist bei Zertifikatsprüfungen relevant.

Ist Datum oder Uhrzeit des Rescue-Systems erheblich falsch, können eigentlich gültige Zertifikate als noch nicht gültig oder bereits abgelaufen erscheinen.

Bei unerwarteten Zertifikatsfehlern gehört die Kontrolle der Systemzeit daher zu den sinnvollen Diagnosemaßnahmen.

Sicherheitsanzeige

Chromium stellt Informationen zum Sicherheitszustand einer Verbindung in der Benutzeroberfläche bereit.

Eine sichere Verbindung bedeutet dabei primär, dass die Verbindung zum angegebenen Ziel entsprechend geschützt wurde.

Sie ist kein Gütesiegel für:

den Inhalt,

den Betreiber,

heruntergeladene Dateien

oder geschäftliche Aussagen der Webseite.

Weiterleitungen

Webserver können den Browser auf eine andere URL weiterleiten.

Typische HTTP-Statuscodes für Redirects sind beispielsweise:

`301`

`302`

`303`

`307`

`308`.

Weiterleitungen können legitim sein, beispielsweise:

HTTP → HTTPS,

alter Hostname → neuer Hostname,

Loginseite → Anwendung.

Sie können bei fehlerhafter Konfiguration aber auch zu:

Redirect-Schleifen,

falschen Zielseiten

oder unerwarteten Hosts

führen.

HTTP-Statuscodes

Chromium beziehungsweise die Developer Tools können HTTP-Statuscodes sichtbar machen.

Wichtige Gruppen sind:

`1xx` – Information

`2xx` – erfolgreiche Verarbeitung

`3xx` – Weiterleitung

`4xx` – clientbezogene Fehler beziehungsweise Anforderungen

`5xx` – serverseitige Fehler.

Beispiele

`200 OK`

zeigt eine erfolgreiche HTTP-Antwort.

`301` oder `302`

zeigen eine Weiterleitung.

`401`

weist auf eine erforderliche beziehungsweise fehlgeschlagene Authentifizierung hin.

`403`

zeigt, dass der Server die Anfrage ablehnt.

`404`

bedeutet, dass die angeforderte Ressource nicht gefunden wurde.

`500`

weist auf einen internen Serverfehler hin.

`502`

kann auf ein Problem zwischen Gateway/Proxy und Backend hinweisen.

`503`

zeigt, dass der Dienst aktuell nicht verfügbar ist.

Ein HTTP-Statuscode muss immer im Kontext der konkreten Anwendung interpretiert werden.

Developer Tools

Chromium enthält integrierte Developer Tools – DevTools.

Diese Werkzeuge sind nicht ausschließlich für Webentwickler interessant.

Im Rescue-Einsatz können sie bei der Untersuchung webbasierter Systeme sehr hilfreich sein.

DevTools öffnen

Unter Linux kann DevTools typischerweise mit:

`Ctrl` + `Shift` + `I`

geöffnet werden.

Alternativ kann die Funktion über das Browsermenü aufgerufen werden.

Network Panel

Für die Rescue-Diagnose ist insbesondere das:

Network Panel

wichtig.

Solange die Developer Tools geöffnet sind und die Aufzeichnung aktiv ist, protokolliert das Network Panel die Netzwerkrequests der Seite.

Was das Network Panel sichtbar macht

Zu einzelnen Requests können unter anderem untersucht werden:

URL,

HTTP-Methode,

Status,

Ressourcentyp,

Initiator,

übertragene Datenmenge,

Zeitverhalten,

Request Headers,

Response Headers,

Request Payload,

Response,

Cookies

und Timing.

Diagnosebeispiel

Eine Appliance-Oberfläche wird angezeigt, aber bestimmte Informationen fehlen.

Im Network Panel kann sichtbar werden:

HTML-Seite: `200`

JavaScript: `200`

API-Aufruf: `500`

Damit wird erkennbar:

Die Weboberfläche selbst wurde geladen.

Das eigentliche Problem liegt möglicherweise bei einer Backend-API.

Request Headers

Request Headers zeigen Informationen, die Chromium an den Server sendet.

Dazu können beispielsweise gehören:

Hostinformationen,

Accept-Header,

Content-Type,

Cookies,

Origin,

Referer

und weitere HTTP-Header.

Nicht jeder Request enthält dieselben Header.

Response Headers

Response Headers stammen vom Server.

Sie können beispielsweise Informationen enthalten über:

Content-Type,

Caching,

Weiterleitungen,

Cookies,

Sicherheitsrichtlinien

und weitere HTTP-Eigenschaften.

Payload

Bei Requests, die Daten übertragen, kann DevTools die Request-Nutzlast anzeigen.

Dies ist diagnostisch hilfreich, kann aber gleichzeitig sensible Daten offenlegen.

Dazu können gehören:

Formulardaten,

API-Parameter,

Benutzernamen,

Tokens

oder andere Anwendungsdaten.

Response

Der Bereich:

Response

zeigt den vom Server zurückgelieferten Inhalt einer ausgewählten Anfrage, soweit DevTools ihn darstellen kann.

Dies ist beispielsweise bei API-Fehlern hilfreich.

Eine grafische Oberfläche kann lediglich:

„Fehler“

anzeigen, während die zugrunde liegende Serverantwort detailliertere technische Informationen enthält.

Initiator

DevTools kann anzeigen, wodurch ein Request ausgelöst wurde.

Dies hilft beispielsweise festzustellen, ob eine Ressource:

durch das HTML-Dokument,

ein Script,

eine Weiterleitung

oder einen anderen Mechanismus

angefordert wurde.

Timing

Der Timing-Bereich zerlegt die zeitlichen Bestandteile eines Requests.

Damit lässt sich genauer untersuchen, wo Zeit innerhalb eines Webrequests verbraucht wird.

Ein langsam wirkender Webdienst muss nicht zwingend durch reine Netzwerklatenz langsam sein.

Auch Serververarbeitung oder andere Bestandteile können erheblichen Anteil haben.

Preserve Log

Mit:

Preserve log

kann das Network Panel Requests über Seitenwechsel beziehungsweise Reloads hinweg erhalten.

Dies ist besonders nützlich bei:

Loginvorgängen,

Redirect-Ketten

und Fehlern, die unmittelbar nach einer Navigation auftreten.

Cache deaktivieren

Während DevTools geöffnet ist, kann für Diagnosezwecke der Browsercache deaktiviert werden.

Damit lässt sich prüfen, ob ein Problem durch zwischengespeicherte Ressourcen beeinflusst wird.

Hard Reload

Für Webdiagnosen kann eine Seite ohne die übliche Cache-Nutzung neu geladen werden.

Dies ist hilfreich, wenn beispielsweise:

alte JavaScript-Dateien,

alte CSS-Dateien

oder andere gecachte Ressourcen

als Fehlerursache vermutet werden.

Filter

Das Network Panel kann Requests filtern.

Damit lassen sich beispielsweise gezielt betrachten:

Fetch/XHR,

JavaScript,

CSS,

Bilder,

Dokumente,

WebSockets

oder bestimmte Domains beziehungsweise Request-Eigenschaften.

Fetch/XHR

Moderne Webanwendungen laden viele Daten nach dem initialen Seitenaufbau über:

Fetch

oder:

XMLHttpRequest.

Eine Seite kann deshalb optisch geladen sein, obwohl die eigentliche Datenkommunikation fehlschlägt.

Für Appliance-Diagnosen sind Fetch-/XHR-Requests daher häufig besonders interessant.

WebSocket

Webanwendungen können dauerhafte bidirektionale WebSocket-Verbindungen verwenden.

DevTools kann entsprechende Verbindungen und Nachrichten untersuchen.

Dies kann bei:

Live-Monitoring,

Statusanzeigen,

Dashboards

und administrativen Webanwendungen

hilfreich sein.

HAR

Netzwerkrequests können aus DevTools als:

HAR – HTTP Archive

exportiert werden.

Eine HAR-Datei kann zahlreiche Details über Webrequests enthalten.

HAR-Dateien können hochsensibel sein

Je nach Anwendung können HAR-Daten enthalten:

URLs,

Query-Parameter,

Header,

Cookies,

Request-Inhalte,

Antwortinformationen

und andere Sitzungsdaten.

Eine HAR-Datei darf deshalb nicht unkontrolliert weitergegeben werden.

Vor einer Weitergabe muss geprüft werden, ob sie:

Authentifizierungsinformationen,

Session-IDs,

Tokens,

personenbezogene Daten

oder interne Infrastrukturinformationen

enthält.

DevTools ist kein vollständiger Paketmitschnitt

Das Network Panel arbeitet auf Browser- beziehungsweise Webanwendungsebene.

Es ersetzt keinen vollständigen Netzwerk-Capture.

Für eine paketgenaue Analyse dienen innerhalb von JUMBO BLOCK Rescue:

TCPDump

oder:

Wireshark.

Wichtige Trennung

Chromium DevTools

zeigt die Webkommunikation aus Sicht des Browsers.

TCPDump/Wireshark

zeigen den am Capture-Punkt sichtbaren Netzwerkverkehr.

Diese Perspektiven ergänzen sich.

Inkognito-Modus

Chromium besitzt einen Inkognito-Modus.

Inkognito ist ein eigener Browserfenster-Modus.

Die darin besuchten Seiten werden nicht in der normalen Browserhistorie gespeichert.

Die Sitzung verwendet einen temporären Cookie-/Site-Data-Kontext.

Nach Ende der Inkognito-Sitzung werden diese temporären Sitzungsdaten verworfen.

Sinn im Rescue-System

Inkognito kann nützlich sein, um eine Weboberfläche ohne den normalen bestehenden Browserzustand zu testen.

Dies kann helfen, Einflüsse von:

bestehenden Cookies,

Anmeldesitzungen

und bestimmten gespeicherten Seitendaten

zu reduzieren.

Inkognito bedeutet nicht Anonymität

Der Inkognito-Modus macht den Benutzer im Netzwerk nicht unsichtbar.

Webseiten können weiterhin Netzwerkkommunikation sehen.

Auch beispielsweise:

Netzwerkadministratoren,

Internetprovider,

Firewalls,

Proxies

und andere Infrastruktur auf dem Netzwerkpfad

können abhängig von ihrer Position und technischen Möglichkeiten Netzwerkaktivität erkennen.

Downloads im Inkognito-Modus

Heruntergeladene Dateien werden durch das Schließen der Inkognito-Sitzung nicht automatisch gelöscht.

Sie bleiben im Dateisystem vorhanden, bis sie ausdrücklich entfernt werden.

Das ist für ein Rescue-System besonders wichtig.

Lesezeichen im Inkognito-Modus

Auch ausdrücklich angelegte Lesezeichen werden durch das Beenden einer Inkognito-Sitzung nicht automatisch entfernt.

Inkognito und forensische Anforderungen

Inkognito ist kein forensischer Spurenvernichtungsmodus.

Es darf nicht als Garantie interpretiert werden, dass auf dem Gesamtsystem oder außerhalb des Browsers keinerlei Spuren einer Aktivität existieren.

Downloads

Chromium kann Dateien aus dem Internet oder von lokalen Websystemen herunterladen.

Dies ist im Rescue-Betrieb beispielsweise nützlich für:

Dokumentationen,

Konfigurationsdateien,

Logs,

Firmwaredateien,

Treiber,

Prüfsummen

oder andere benötigte Ressourcen.

Download bedeutet nicht Vertrauenswürdigkeit

Eine erfolgreich über HTTPS heruntergeladene Datei ist nicht automatisch sicher.

HTTPS schützt die Übertragung zur authentifizierten Gegenstelle.

Es beweist nicht, dass die Datei:

schadcodefrei,

unverändert gegenüber einer Herstellerreferenz,

für das Zielsystem geeignet

oder tatsächlich vom erwarteten Hersteller erstellt

wurde.

Downloads verifizieren

Bei sicherheitsrelevanten Downloads sollten – sofern vom Hersteller bereitgestellt – zusätzlich geprüft werden:

Herkunft,

Dateiname,

Version,

kryptografische Prüfsumme

und/oder:

digitale Signatur.

Innerhalb von JUMBO BLOCK Rescue können hierfür je nach Dateityp andere Werkzeuge ergänzend eingesetzt werden.

Download aus unbekannter Quelle

Dateien unbekannter Herkunft sollten nicht unmittelbar ausgeführt oder geöffnet werden.

Das gilt insbesondere für:

Programme,

Scripts,

Archive,

Office-Dokumente,

PDF-Dateien

und andere aktive beziehungsweise komplexe Dateiformate.

ClamAV als Ergänzung

Eine heruntergeladene Datei kann vor weiterer Verwendung mit ClamAV untersucht werden.

Dies erhöht die diagnostische Sicherheit, ersetzt aber keine:

Herkunftsprüfung,

Signaturprüfung

oder manuelle Bewertung.

Ein fehlender ClamAV-Fund beweist nicht, dass eine Datei ungefährlich ist.

GnuPG als Ergänzung

Stellt ein Hersteller eine OpenPGP-Signatur bereit, kann später im Verschlüsselungsblock dokumentiertes GnuPG zur Signaturprüfung eingesetzt werden.

Damit lässt sich – bei korrekt geprüfter Schlüsselkette – die Authentizität beziehungsweise Integrität deutlich stärker bewerten als allein anhand eines Browserdownloads.

Sandbox

Chromium verwendet eine Mehrprozessarchitektur und Sandboxmechanismen.

Webinhalte werden nicht einfach mit uneingeschränkten Systemrechten in einem einzigen Browserprozess ausgeführt.

Rendererprozesse werden eingeschränkt, um die Auswirkungen einer kompromittierten Rendering-Komponente zu begrenzen.

Sandbox ist keine absolute Sicherheitsgrenze

Sandboxing reduziert Risiken.

Es bedeutet nicht:

„Eine beliebige schädliche Webseite kann gefahrlos geöffnet werden.“

Browser und Sandbox können Sicherheitslücken besitzen.

Der sichere Grundsatz bleibt deshalb:

Unbekannte oder offensichtlich verdächtige Inhalte nur öffnen, wenn dies für die Diagnose notwendig und vertretbar ist.

`chromium-sandbox`

Die Baseline enthält ausdrücklich das Paket:

`chromium-sandbox 154.0.8037.57-1~deb13u1`.

Damit ist die zu dieser Chromium-Paketversion gehörende Sandbox-Komponente als installiert belegt.

Aus der Paketliste allein wird jedoch nicht abgeleitet, welcher konkrete Sandboxmechanismus für jeden einzelnen Chromium-Prozess zur Laufzeit aktiv ist.

Site Isolation

Chromium verwendet auf Desktop-Systemen Site Isolation als zusätzliche Sicherheitsarchitektur.

Webseiten verschiedener Sites werden in getrennten Rendererprozessen isoliert.

Damit soll insbesondere verhindert beziehungsweise erschwert werden, dass eine kompromittierte Webseite Daten anderer Sites aus einem gemeinsamen Rendererprozess auslesen kann.

Defense in Depth

Sandbox und Site Isolation sind Beispiele für:

Defense in Depth.

Die Sicherheitsarchitektur verlässt sich nicht ausschließlich darauf, dass Webinhalte fehlerfrei verarbeitet werden.

Zusätzliche technische Grenzen sollen die Auswirkungen eines Fehlers reduzieren.

Kein Grund, Sicherheitsmechanismen abzuschalten

Für einen Rescue-Browser wäre es sicherheitstechnisch besonders problematisch, Schutzmechanismen wie die Sandbox ohne zwingenden technischen Grund zu deaktivieren.

Ein Start mit Optionen, die Browser-Sicherheitsgrenzen abschalten, sollte daher nicht Bestandteil eines normalen Rescue-Workflows sein.

Lokale Geräte mit alten Weboberflächen

Ältere Appliances können Browserkompatibilitätsprobleme verursachen.

Beispiele sind:

veraltete TLS-Konfigurationen,

nicht mehr unterstützte Kryptografie,

alte JavaScript-Implementierungen,

unsichere Zertifikate

oder proprietäre Browseranforderungen.

Sicherheitsgrenzen nicht für Kompatibilität opfern

Wenn eine alte Appliance mit aktuellem Chromium nicht mehr funktioniert, bedeutet dies nicht automatisch, dass Chromium fehlerhaft ist.

Der Grund kann eine veraltete Sicherheitskonfiguration des Geräts sein.

Ein Rescue-System sollte nicht leichtfertig seine Browser-Sicherheitsmechanismen reduzieren, nur um eine technisch überholte Weboberfläche erreichbar zu machen.

Captive Portals

Bei WLAN- oder anderen Netzwerken kann vor dem eigentlichen Internetzugriff ein Captive Portal erforderlich sein.

Ein Browser ist für diese Anmeldung häufig notwendig.

Ein Zertifikats- oder Redirectproblem direkt nach einer Netzwerkverbindung kann deshalb auch mit einem Captive Portal zusammenhängen.

Proxy

Webzugriffe können über einen Proxy geleitet werden.

Ein Browserproblem kann deshalb entstehen, obwohl:

DNS,

IP-Routing

und direkter Netzwerkzugriff

grundsätzlich funktionieren.

Bei unerwarteten Browserfehlern gehört eine mögliche Proxykonfiguration daher zur Diagnose.

Cookies

Cookies speichern kleine Datenobjekte im Browserkontext.

Sie werden häufig für:

Sitzungen,

Anmeldestatus,

Präferenzen

und Webanwendungsfunktionen

verwendet.

Fehlerhafte oder veraltete Cookies können eine Webanwendung beeinflussen.

Cookies können sicherheitsrelevant sein

Session-Cookies können Authentifizierungsinformationen repräsentieren.

Sie dürfen nicht unkontrolliert:

exportiert,

protokolliert,

weitergegeben

oder in Screenshots beziehungsweise HAR-Dateien veröffentlicht

werden.

Local Storage und andere Site-Daten

Moderne Webanwendungen verwenden neben Cookies weitere lokale Browsermechanismen.

Dadurch kann ein Fehler auch von gespeichertem Browserzustand abhängen.

Ein Test in einer frischen Inkognito-Sitzung kann helfen festzustellen, ob der bestehende Browserzustand Einfluss hat.

Cache

Chromium speichert Webressourcen zwischen, um wiederholte Zugriffe zu beschleunigen.

Bei administrativen Weboberflächen kann ein veralteter Cache jedoch gelegentlich zu einer Mischung aus alten und neuen Webressourcen führen.

DevTools erlaubt deshalb gezielte Tests ohne Cache.

JavaScript

Viele moderne Verwaltungsoberflächen sind stark von JavaScript abhängig.

Wenn JavaScript-Code nicht geladen wird oder während der Ausführung fehlschlägt, kann eine Seite:

leer bleiben,

nur teilweise erscheinen,

keine Daten anzeigen

oder Bedienelemente nicht ausführen.

Console

Die DevTools-Konsole kann JavaScript-Fehler und andere Meldungen der Webseite anzeigen.

Sie ist deshalb eine wichtige Ergänzung zum Network Panel.

Vorsicht mit Console-Befehlen

Code aus unbekannten Webseiten, Foren oder Nachrichten sollte nicht unkritisch in die DevTools-Konsole kopiert werden.

JavaScript, das dort manuell ausgeführt wird, arbeitet im Kontext der betreffenden Webseite und kann sicherheitsrelevante Aktionen auslösen.

Passwort- und Zugangsdaten

Auf einem Rescue-System sollte bewusst entschieden werden, ob Zugangsdaten im Browser gespeichert werden.

Bei einem temporären oder gemeinsam verwendeten Rescue-System ist dauerhafte Speicherung von:

Passwörtern,

Tokens,

Formulardaten

oder Sitzungsinformationen

besonders kritisch.

Rescue-System als temporäre Arbeitsumgebung

Wird JUMBO BLOCK Rescue als Live-/Rescue-System eingesetzt, sollte nicht automatisch davon ausgegangen werden, dass jeder Browserzustand nach dem Neustart verschwindet.

Dies hängt von:

Bootmedium,

Persistenz,

Mounts,

Downloadzielen

und konkreter Systemkonfiguration

ab.

Aus der Chromium-Paketbaseline allein lässt sich keine Aussage über das Persistenzverhalten von JUMBO BLOCK Rescue ableiten.

Typischer Rescue-Ablauf: lokale Routeroberfläche nicht erreichbar

1. Lokale Netzkonfiguration prüfen

Ist das Rescue-System im richtigen Netz?

2. ARP Scan einsetzen

Ist das Gerät auf dem lokalen IPv4-Layer-2-Netz sichtbar?

3. Nmap einsetzen

Ist der erwartete HTTP-/HTTPS-Port erreichbar?

4. Chromium öffnen

Direkte Geräteoberfläche aufrufen.

5. Browserfehler genau dokumentieren

DNS?

Timeout?

Connection refused?

TLS?

Zertifikat?

HTTP-Fehler?

6. DevTools Network öffnen

HTTP-Kommunikation untersuchen.

7. Bei Bedarf TCPDump oder Wireshark ergänzen

Paketebene untersuchen.

Typischer Rescue-Ablauf: Hostname funktioniert nicht

1. Chromium-Fehler dokumentieren

2. Dig verwenden

A-/AAAA-Auflösung prüfen.

3. Gegebenenfalls direkte IP-Verbindung testen

4. Zertifikats- und Hostname-Unterschied berücksichtigen

5. MTR für den Netzwerkpfad einsetzen

6. Nmap für den Zielport einsetzen

Damit wird die Browsermeldung in einzelne Diagnoseebenen zerlegt.

Typischer Rescue-Ablauf: Webseite lädt teilweise

1. DevTools öffnen

2. Network Panel aktivieren

3. Seite neu laden

4. Fehlgeschlagene Requests identifizieren

Besonders auf:

4xx,

5xx,

CORS-Fehler,

fehlgeschlagene Requests

und ungewöhnlich lange Requests

achten.

5. Request auswählen

Headers, Response und Timing untersuchen.

6. Console prüfen

JavaScript- beziehungsweise Webanwendungsfehler kontrollieren.

7. Bei Bedarf ohne Cache wiederholen

Typischer Rescue-Ablauf: HTTPS-Warnung bei lokaler Appliance

1. IP-Adresse und Hostname kontrollieren

2. Sicherstellen, dass das erwartete Gerät angesprochen wird

3. Systemzeit prüfen

4. Zertifikatsfehler genau lesen

5. Zertifikatsdaten untersuchen

6. Mit bekannter Gerätedokumentation vergleichen

7. Nur nach bewusster Risikobewertung fortfahren

Eine Zertifikatswarnung ist kein lästiger Dialog, sondern ein Hinweis darauf, dass die erwartete Authentifizierung der Gegenstelle nicht vollständig bestätigt werden konnte.

Typischer Rescue-Ablauf: Download einer Reparaturdatei

1. Offizielle Quelle bestimmen

2. HTTPS-Verbindung und Zielhostname prüfen

3. Datei herunterladen

4. Hersteller-Prüfsumme beziehungsweise Signatur beschaffen

5. Datei kryptografisch verifizieren

6. Gegebenenfalls zusätzlich mit ClamAV untersuchen

7. Erst danach verwenden

Chromium und Dig

Dig untersucht DNS direkt.

Chromium verwendet Namensauflösung als Teil des Webzugriffs.

Wenn Chromium einen Hostnamen nicht erreichen kann, hilft Dig bei der Trennung:

DNS-Problem

oder:

Browser-/Webproblem.

Chromium und MTR

MTR untersucht den Netzwerkpfad.

Chromium untersucht die eigentliche Webverbindung.

Eine langsame Webseite kann durch:

Netzwerklatenz,

Paketverlust,

Serververarbeitung,

viele Webrequests,

JavaScript

oder andere Faktoren

verursacht werden.

MTR und Chromium DevTools liefern deshalb unterschiedliche Perspektiven.

Chromium und Nmap

Nmap kann feststellen, ob beispielsweise TCP-Port 80 oder 443 an einem autorisierten Ziel erreichbar erscheint.

Chromium prüft anschließend, ob dort tatsächlich eine für den Browser nutzbare Webanwendung bereitsteht.

Vereinfacht:

Nmap → Ist der Port/Dienst netzwerkseitig sichtbar?

Chromium → Funktioniert die Webanwendung über diesen Dienst?

Chromium und TCPDump

TCPDump kann die durch Chromium erzeugte Netzwerkkommunikation terminalbasiert aufzeichnen.

Dies ist hilfreich, wenn Browsermeldungen allein nicht erklären, was auf dem Netzwerk passiert.

Chromium und Wireshark

Wireshark bietet die detaillierteste grafische Paketperspektive.

Chromium DevTools kennt dagegen die Webanwendungsstruktur und kann beispielsweise:

URLs,

HTTP-Requests,

Header,

Statuscodes,

Initiatoren

und Webanwendungs-Timing

direkt zuordnen.

Beide Werkzeuge ergänzen sich.

Chromium und Whois

Eine über Chromium aufgerufene öffentliche Domain kann mit Dig aufgelöst werden.

Die resultierende öffentliche IP-Adresse kann anschließend mit Whois registryseitig eingeordnet werden.

Damit ergibt sich:

Chromium → Webanwendung

Dig → DNS

Whois → Registryzuordnung

Grenzen

Chromium ist kein Paket-Sniffer.

Chromium ersetzt Wireshark oder TCPDump nicht.

Chromium ist kein Portscanner.

Chromium ersetzt Nmap nicht.

Chromium ist kein DNS-Diagnosewerkzeug auf Protokollebene.

Chromium ersetzt Dig nicht.

Eine funktionierende Webseite beweist nicht, dass alle Netzwerkdienste des Systems funktionieren.

Eine HTTPS-Verbindung beweist nicht, dass der Inhalt einer Webseite vertrauenswürdig ist.

Eine Zertifikatswarnung darf nicht automatisch ignoriert werden.

Inkognito macht den Benutzer nicht anonym.

Inkognito löscht heruntergeladene Dateien nicht automatisch.

Die Sandbox reduziert Risiken, macht das Öffnen unbekannter Inhalte aber nicht risikofrei.

DevTools zeigt die Browserperspektive und keinen vollständigen Paketmitschnitt.

Sicherheitshinweise

Chromium in einer Rescue-Umgebung grundsätzlich als aktiven Netzwerkclient betrachten.

Unbekannte und kompromittierungsverdächtige Webseiten nur öffnen, wenn dies für die Diagnose erforderlich ist.

Zertifikatswarnungen nicht routinemäßig umgehen.

Vor der Eingabe von Zugangsdaten Hostname und Verbindungszustand prüfen.

Downloads unbekannter Herkunft nicht unmittelbar ausführen.

Herstellerdateien nach Möglichkeit über kryptografische Prüfsummen oder digitale Signaturen verifizieren.

Sensible Downloads gegebenenfalls zusätzlich mit ClamAV untersuchen.

Keine unbekannten Scripts in die DevTools-Konsole kopieren.

HAR-Dateien vor Weitergabe auf Zugangsdaten, Cookies und Tokens prüfen.

Session-Cookies und Authentifizierungsinformationen vertraulich behandeln.

Bei temporären Rescue-Systemen nicht ohne Prüfung annehmen, dass alle Browserdaten beim Neustart verschwinden.

Sandbox- und andere Browser-Sicherheitsmechanismen nicht leichtfertig deaktivieren.

Technische Einordnung in JUMBO BLOCK Rescue

Chromium übernimmt innerhalb des Netzwerkbereichs von JUMBO BLOCK Rescue die Rolle des **grafischen Webclients und der Webanwendungsdiagnose**.

Damit ist Block A.17 bis A.24 funktional vollständig gegliedert:

A.17 Nmap

aktive Host-, Port- und Dienstanalyse.

A.18 Wireshark

grafische Paket- und Protokollanalyse.

A.19 TCPDump

terminalbasierte Paketaufzeichnung und Filterung.

A.20 ARP Scan

lokale IPv4-Geräteerkennung auf Layer 2.

A.21 MTR

fortlaufende Pfad-, Latenz- und Paketverlustanalyse.

A.22 Dig

DNS-, Resolver-, Resource-Record- und Delegationsanalyse.

A.23 Whois

Registry-, Netzblock-, Domain- und AS-Einordnung.

A.24 Chromium

grafischer Webzugriff sowie HTTP-/HTTPS- und Webanwendungsdiagnose.

Die Werkzeuge bilden damit eine Diagnosekette über mehrere Ebenen:

ARP Scan

lokales Gerät erkennen.

↓

Dig

Namensauflösung untersuchen.

↓

MTR

Netzwerkpfad analysieren.

↓

Nmap

Zielports und Dienste untersuchen.

↓

Chromium

Webdienst und Benutzeroberfläche prüfen.

↓

DevTools

HTTP-/Webanwendungskommunikation analysieren.

↓

TCPDump / Wireshark

bei Bedarf auf die Paketebene wechseln.

↓

Whois

öffentliche Netzwerkobjekte administrativ einordnen.

Chromium nimmt dabei eine besondere Stellung ein:

Es ist nicht nur ein Werkzeug für den Zugriff auf das Internet.

In einem modernen Rescue-System ist der Browser zugleich die **Bedien- und Diagnoseoberfläche für eine große Zahl webbasierter Infrastrukturkomponenten**.

Seine wichtigste diagnostische Stärke liegt deshalb in der Verbindung von:

realem Webzugriff,

HTTP-/HTTPS-Verhalten,

TLS-/Zertifikatsprüfung

und:

integrierter Request-Analyse über DevTools.

Für den Rescue-Einsatz gilt dabei als zentrale Sicherheitsregel:

Eine Browserwarnung sollte nicht beseitigt werden, bevor verstanden wurde, warum sie erscheint.

A.25 Cryptsetup

Bereich: Verschlüsselung
Programm: Cryptsetup
Baseline 20261001: Cryptsetup 2.7.5
Installierte Pakete: `cryptsetup 2:2.7.5-2`, `cryptsetup-bin 2:2.7.5-2`, `cryptsetup-initramfs 2:2.7.5-2`
Bedienung: Kommandozeile
Aufgabe: Erkennung, Analyse, Öffnung und Verwaltung verschlüsselter Datenträger, insbesondere LUKS-Volumes

Zweck

Cryptsetup ist das zentrale Linux-Kommandozeilenwerkzeug zur Verwaltung verschlüsselter Blockgeräte auf Basis des Kernel-Subsystems:

dm-crypt

und des darauf aufbauenden Formats:

LUKS – Linux Unified Key Setup.

Cryptsetup kann verschlüsselte Datenträger beziehungsweise Partitionen:

erkennen,

Metadaten untersuchen,

entsperren,

als Device-Mapper-Gerät bereitstellen,

wieder schließen

und – bei administrativer Verwendung – Verschlüsselungsheader, Schlüssel und Keyslots verwalten.

Für JUMBO BLOCK Rescue steht primär die sichere **Analyse und kontrollierte Öffnung bereits vorhandener verschlüsselter Datenträger** im Vordergrund.

Einsatzgebiet im Rescue-System

Cryptsetup eignet sich innerhalb von JUMBO BLOCK Rescue insbesondere für:

Erkennung von LUKS-Volumes,

Untersuchung von LUKS-Metadaten,

Ermittlung von LUKS-Version und UUID,

Anzeige vorhandener Keyslots,

kontrolliertes Entsperren verschlüsselter Partitionen,

Bereitstellung entschlüsselter Blockgeräte über Device Mapper,

Read-only-orientierte Datenrettungsabläufe,

Überprüfung aktiver Cryptsetup-Mappings,

Schließen entschlüsselter Mappings,

Sicherung eines LUKS-Headers

und – nur bei ausdrücklich erforderlichen administrativen Arbeiten – Verwaltung von Passphrasen und Keyslots.

Cryptsetup und dm-crypt

Cryptsetup führt die eigentliche Blockverschlüsselung nicht allein im Benutzerprozess durch.

Es konfiguriert unter Linux unter anderem das Kernel-Subsystem:

`dm-crypt`.

Ein erfolgreich geöffnetes verschlüsseltes Volume erscheint typischerweise als Device-Mapper-Gerät unter:

`/dev/mapper/`

Dieses entschlüsselte Blockgerät kann anschließend wie ein normales Blockgerät weiter untersucht werden.

Wichtige Ebenentrennung

Bei einem LUKS-Datenträger müssen mehrere Ebenen unterschieden werden:

Physisches beziehungsweise ursprüngliches Blockgerät

zum Beispiel eine Partition.

↓

LUKS-Verschlüsselungsschicht

Cryptsetup entschlüsselt die Daten blockweise.

↓

Device-Mapper-Mapping

zum Beispiel:

`/dev/mapper/rescue`

↓

Dateisystem

zum Beispiel ext4, XFS oder ein anderes unterstütztes Dateisystem.

↓

Mountpunkt

erst hier werden Dateien und Verzeichnisse sichtbar.

Diese Ebenen dürfen in der Rescue-Diagnose nicht miteinander verwechselt werden.

LUKS

LUKS ist ein standardisiertes Linux-Format für Blockgeräteverschlüsselung.

Im Gegensatz zu einfachem Plain-dm-crypt enthält ein LUKS-Datenträger Metadaten zur Verwaltung der Verschlüsselung.

Dazu gehören unter anderem Informationen über:

Formatversion,

Verschlüsselungsparameter,

UUID,

Keyslots

und die für die Schlüsselableitung erforderlichen Parameter.

LUKS1 und LUKS2

Cryptsetup 2.7.5 unterstützt sowohl:

LUKS1

als auch:

LUKS2.

LUKS2 besitzt ein erweitertes Metadatenformat und zusätzliche Verwaltungsfunktionen.

Bei Rescue-Arbeiten ist deshalb zunächst festzustellen, welcher LUKS-Typ tatsächlich vorliegt.

LUKS erkennen

Eine besonders wichtige nicht destruktive Prüfung lautet:

```
cryptsetup isLuks DEVICE
```

Der Befehl prüft, ob Cryptsetup auf dem angegebenen Gerät einen gültigen LUKS-Header erkennt.

Ausführliche Prüfung

Mit:

```
cryptsetup -v isLuks DEVICE
```

wird zusätzlich eine textuelle Ausgabe erzeugt.

Ohne `-v` wird das Ergebnis im Wesentlichen über den Exitstatus signalisiert.

Bedeutung von `isLuks`

Ein erfolgreiches:

`isLuks`

bedeutet:

Auf dem untersuchten Gerät wurde ein von Cryptsetup als LUKS erkannter Header gefunden.

Es bedeutet nicht:

dass eine bekannte Passphrase vorhanden ist,

dass das Dateisystem innerhalb des Containers intakt ist,

oder:

dass sämtliche verschlüsselten Nutzdaten fehlerfrei lesbar sind.

Keine LUKS-Erkennung

Schlägt `isLuks` fehl, kann dies verschiedene Ursachen haben.

Beispielsweise:

Gerät ist nicht LUKS-formatiert,

falsches Blockgerät untersucht,

LUKS-Header beschädigt,

LUKS-Header liegt separat vor,

oder das Gerät verwendet ein anderes Verschlüsselungsformat.

Aus einem negativen `isLuks`-Ergebnis darf daher bei einem beschädigten Datenträger nicht automatisch geschlossen werden, dass niemals LUKS vorhanden war.

Header untersuchen

Für einen erkannten LUKS-Datenträger ist:

`cryptsetup luksDump DEVICE`

eines der wichtigsten Diagnosekommandos.

`luksDump` zeigt die LUKS-Headerinformationen an.

`luksDump`

Abhängig von LUKS-Version und Konfiguration können dabei unter anderem Informationen sichtbar werden über:

LUKS-Version,

UUID,

Verschlüsselungsparameter,

Keyslots,

Key-Derivation-Parameter

und weitere Metadaten.

`luksDump` entschlüsselt nicht die Nutzdaten

Der Befehl:

```
cryptsetup luksDump DEVICE
```

öffnet nicht automatisch das Dateisystem und entschlüsselt nicht den gesamten Datenträger.

Er untersucht die im LUKS-Header vorhandenen Metadaten.

Damit eignet er sich besonders gut für eine erste Bestandsaufnahme.

UUID

Mit:

```
cryptsetup luksUUID DEVICE
```

kann die UUID eines LUKS-Geräts ausgegeben werden.

Diese UUID identifiziert das LUKS-Volume.

UUID nicht mit Dateisystem-UUID verwechseln

Ein entschlüsseltes Dateisystem kann seinerseits eine eigene UUID besitzen.

Deshalb muss unterschieden werden:

LUKS UUID

und:

Dateisystem-UUID innerhalb des entschlüsselten Volumes.

Sie gehören zu unterschiedlichen Ebenen.

Keyslots

LUKS kann mehrere Schlüsselzugänge beziehungsweise Keyslots verwalten.

Verschiedene Passphrasen können dadurch Zugriff auf denselben zugrunde liegenden Volume Key ermöglichen.

Passphrase ist nicht direkt der Datenverschlüsselungsschlüssel

Bei LUKS wird eine Benutzerpassphrase nicht einfach unmittelbar als Schlüssel für jeden Datenblock verwendet.

Die Passphrase dient dazu, über die LUKS-Schlüsselverwaltung Zugriff auf den eigentlichen Volume Key zu erhalten.

Die Datenverschlüsselung erfolgt mit diesem Volume Key.

Konsequenz

Mehrere unterschiedliche Passphrasen können dasselbe verschlüsselte Volume öffnen.

Das Ändern einer Passphrase bedeutet daher nicht, dass sämtliche Nutzdaten mit einem neuen Volume Key vollständig neu verschlüsselt werden.

LUKS-Volume öffnen

Ein LUKS-Gerät kann mit:

```
cryptsetup open DEVICE NAME
```

geöffnet werden.

Für LUKS kann der Typ ausdrücklich angegeben werden:

```
cryptsetup open --type luks DEVICE NAME
```

Die traditionelle LUKS-Syntax lautet:

```
cryptsetup luksOpen DEVICE NAME
```

Nach erfolgreicher Authentifizierung erzeugt Cryptsetup ein Mapping mit dem angegebenen Namen.

Beispielprinzip

```
sudo cryptsetup open --type luks /dev/sdXN rescue
```

Nach erfolgreicher Öffnung steht typischerweise ein Gerät zur Verfügung:

```
/dev/mapper/rescue
```

Die tatsächliche Gerätebezeichnung muss vor jedem Rescue-Eingriff eindeutig festgestellt werden.

Mapping-Name

Der Name:

```
rescue
```

ist frei gewählt.

Er bezeichnet nicht die ursprüngliche Partition, sondern das neu erzeugte Device-Mapper-Mapping.

Passphrase

Wird keine andere Schlüsselquelle angegeben, fordert Cryptsetup bei einem normalen interaktiven LUKS-Open typischerweise eine Passphrase an.

Die eingegebene Passphrase wird nicht als Klartext auf der normalen Terminalausgabe dargestellt.

Falsche Passphrase

Kann mit einer eingegebenen Passphrase kein geeigneter Keyslot erfolgreich geöffnet werden, wird kein nutzbares Mapping erzeugt.

Eine falsche Passphrase beschädigt den LUKS-Datenträger nicht.

Read-only öffnen

Für Rescue- und Analysearbeiten ist besonders:

`--readonly`

relevant.

Beispiel:

```
sudo cryptsetup open --readonly --type luks DEVICE rescue
```

Damit wird das erzeugte Mapping als read-only eingerichtet.

Warum Read-only wichtig ist

Bei einem beschädigten oder forensisch sensiblen Datenträger sollte grundsätzlich vermieden werden, unnötige Schreibzugriffe auf das Original zu erzeugen.

Ein read-only geöffnetes Cryptsetup-Mapping reduziert dieses Risiko auf der Verschlüsselungsebene.

Read-only allein genügt nicht immer für einen vollständigen Schutz

Auch die nachfolgenden Ebenen müssen berücksichtigt werden.

Wird beispielsweise ein Dateisystem später auf eine Weise eingebunden, die Schreibzugriffe voraussetzt, kann dies fehlschlagen oder unerwünschte Nebeneffekte erzeugen.

Für einen kontrollierten Rescue-Workflow sollten daher sowohl:

Cryptsetup-Mapping

als auch:

Dateisystemzugriff

bewusst read-only geplant werden.

Geöffnetes Mapping prüfen

Mit:

```
cryptsetup status NAME
```

kann der Status eines aktiven Cryptsetup-Mappings untersucht werden.

Beispiel:

```
cryptsetup status rescue
```

Die Ausgabe kann Informationen über das aktive Mapping und dessen Verschlüsselungskonfiguration enthalten.

Status eines nicht aktiven Mappings

Ist der angegebene Mapping-Name nicht aktiv, meldet Cryptsetup entsprechend, dass kein aktives Gerät dieses Namens vorhanden ist.

Nach dem Öffnen

Nach erfolgreichem:

```
cryptsetup open
```

ist zunächst nur die Verschlüsselungsschicht geöffnet.

Das bedeutet noch nicht, dass das darin enthaltene Dateisystem automatisch eingehängt ist.

Dateisystem untersuchen

Das entschlüsselte Mapping kann anschließend beispielsweise mit geeigneten Systemwerkzeugen auf:

Partitionierung,

Dateisystemtyp,

Dateisystemzustand

oder enthaltene Daten

untersucht werden.

Welches Werkzeug dafür geeignet ist, hängt von der tatsächlichen Struktur innerhalb des entschlüsselten Geräts ab.

Verschlüsselung und Dateisystemschaden unterscheiden

Ein wichtiger Rescue-Fall lautet:

LUKS lässt sich erfolgreich öffnen,

aber das Dateisystem kann nicht gemountet werden.

Dann funktioniert die Entschlüsselung grundsätzlich.

Das Problem kann auf der darüberliegenden Ebene liegen:

Dateisystem beschädigt,

innere Partitionstabelle beschädigt,

RAID/LVM-Struktur vorhanden,

oder anderer Aufbau als erwartet.

LUKS erfolgreich öffnen bedeutet nicht „Daten gesund“

Cryptsetup beantwortet primär:

Kann die Verschlüsselungsschicht geöffnet werden?

Es beantwortet nicht:

Ist das Dateisystem fehlerfrei?

und auch nicht:

Sind alle Dateien unbeschädigt?

Mapping schließen

Ein geöffnetes Mapping wird mit:

```
cryptsetup close NAME
```

wieder geschlossen.

Beispiel:

```
sudo cryptsetup close rescue
```

Die ältere kompatible Aktionsbezeichnung:

```
luksClose
```

ist ebenfalls Teil der Cryptsetup-Befehlssyntax.

Vor dem Schließen

Ein darüber eingehängtes Dateisystem muss zuerst ordnungsgemäß ausgehängt werden.

Die Reihenfolge lautet daher grundsätzlich:

Dateisystem unmounten

↓

Cryptsetup-Mapping schließen

↓

Datenträger gegebenenfalls sicher entfernen

Warum die Reihenfolge wichtig ist

Solange ein Dateisystem auf:

`/dev/mapper/NAME`

eingehängt ist, wird das Mapping noch verwendet.

Ein erzwungenes oder unsauberes Entfernen von Ebenen kann zu Fehlern beziehungsweise Datenverlust führen.

Rescue-Grundablauf

Für einen intakten verschlüsselten Datenträger kann ein konservativer Ablauf lauten:

1. Datenträger eindeutig identifizieren

2. Hardwarezustand berücksichtigen

3. Bei physischem Schaden zuerst Image beziehungsweise Klon erstellen

4. LUKS-Erkennung durchführen

```
cryptsetup -v isLuks DEVICE
```

5. Headerinformationen untersuchen

```
cryptsetup luksDump DEVICE
```

6. Volume möglichst read-only öffnen

```
cryptsetup open --readonly --type luks DEVICE rescue
```

7. Mapping prüfen

```
cryptsetup status rescue
```

8. Dateisystemebene separat untersuchen

9. Daten auf ein anderes Medium sichern

10. Dateisystem aushängen

11. Mapping schließen

```
cryptsetup close rescue
```

Beschädigter Datenträger

Bei einem physisch instabilen Datenträger sollte Cryptsetup nicht das erste Rettungswerkzeug sein.

Wiederholte Leseversuche können einen sterbenden Datenträger zusätzlich belasten.

Hier gilt weiterhin die bereits im Datenrettungsblock dokumentierte Strategie:

Original möglichst wenig belasten → mit GNU ddrescue ein Image oder einen Klon erstellen → anschließend auf der Kopie arbeiten.

Verschlüsseltes Image

Ein sektorweises Image einer LUKS-Partition enthält die verschlüsselten Sektoren einschließlich der für das LUKS-Format relevanten Strukturen.

Ein solches Image bleibt verschlüsselt.

Vorteil

Damit kann die weitere Analyse auf einer Kopie erfolgen, während das Original geschützt bleibt.

Gerade bei:

defekten SSDs,

Festplatten mit Lesefehlern,

USB-Medien mit Aussetzern

oder anderen instabilen Datenträgern

ist dies ein zentraler Rescue-Grundsatz.

Verschlüsselung verhindert kein Imaging

Ein Blockgerät muss nicht zuerst entschlüsselt werden, um ein sektorweises Image davon zu erstellen.

GNU ddrescue arbeitet auf Blockebene und kann die verschlüsselten Sektoren kopieren.

Cryptsetup kann anschließend auf das Image beziehungsweise das daraus bereitgestellte Blockgerät angewendet werden.

Header als kritische Struktur

Der LUKS-Header und die Keyslot-Daten sind für den Zugriff auf verschlüsselte Daten von entscheidender Bedeutung.

Die Cryptsetup-Projektdokumentation warnt ausdrücklich davor, dass Beschädigungen dieser Strukturen zu dauerhaftem Datenverlust führen können.

LUKS1

Bei LUKS1 ist der Header besonders kritisch.

Werden wesentliche Header- beziehungsweise Keyslotinformationen zerstört und existiert kein passendes Backup, können die verschlüsselten Daten kryptografisch nicht mehr zugänglich sein.

LUKS2

LUKS2 besitzt gegenüber LUKS1 zusätzliche Redundanz bei den Metadaten.

Auch LUKS2 ist jedoch nicht gegen jede Beschädigung geschützt.

Ein Header beziehungsweise Keyslot kann weiterhin dauerhaft zerstört werden.

Header-Backup

Cryptsetup besitzt dafür:

`luksHeaderBackup`

Syntax:

```
cryptsetup luksHeaderBackup DEVICE --header-backup-file DATEI
```

Damit wird ein binäres Backup des LUKS-Headers und des Keyslot-Bereichs erzeugt.

Bedeutung des Header-Backups

Ein Header-Backup kann im Katastrophenfall entscheidend sein.

Es ist jedoch selbst hochsensibles Sicherheitsmaterial.

Header-Backup enthält Schlüsselzugangsdaten

Die Cryptsetup-Dokumentation weist ausdrücklich darauf hin:

Ein Header-Backup zusammen mit einer Passphrase, die zum Zeitpunkt des Backups gültig war, kann den Zugriff auf den verschlüsselten Datenbereich ermöglichen – selbst wenn diese Passphrase später auf dem eigentlichen Datenträger geändert oder entfernt wurde.

Konsequenz

Ein LUKS-Header-Backup muss ähnlich geschützt werden wie anderes hochsensibles Schlüsselmaterial.

Es sollte nicht:

öffentlich gespeichert,

ungeschützt per E-Mail versendet,

auf beliebigen Cloudspeichern abgelegt

oder zusammen mit der Passphrase unsicher verwahrt

werden.

Header-Backup verändert die Löschstrategie

Existieren Header-Backups, genügt das alleinige Zerstören der Header-/Keyslot-Strukturen auf dem ursprünglichen Datenträger nicht mehr zwingend, um den Zugriff über diese alten Schlüsselzustände auszuschließen.

Auch vorhandene Header-Backups müssen bei einer sicheren Außerbetriebnahme berücksichtigt werden.

Header wiederherstellen

Die Gegenoperation lautet:

```
cryptsetup luksHeaderRestore DEVICE --header-backup-file DATEI
```

Sie stellt einen gesicherten LUKS-Header und Keyslot-Bereich wieder her.

Höchste Vorsicht bei `luksHeaderRestore`

`luksHeaderRestore` ist eine **schreibende und potenziell folgenschwere Operation**.

Die vorhandenen Header-/Keyslotinformationen werden durch den Backupzustand ersetzt.

Danach gelten die Schlüsselzugänge des wiederhergestellten Backups.

Rescue-Regel

Ein Header-Restore sollte niemals der erste Versuch bei einem unbekannten oder beschädigten LUKS-Datenträger sein.

Vor einem Restore sollten mindestens:

Originalzustand gesichert,

aktueller Header soweit möglich gesichert,

Geräteidentität eindeutig geprüft

und Kompatibilität des Backups festgestellt

werden.

Header testen, ohne ihn sofort zurückzuschreiben

Die Cryptsetup-FAQ beschreibt die Möglichkeit, ein vorhandenes Header-Backup über:

`--header`

als separaten Header zu verwenden.

Dadurch kann unter geeigneten Bedingungen geprüft werden, ob ein Backup das Volume öffnen kann, ohne es unmittelbar auf das Originalgerät zurückzuschreiben.

Dies ist für Rescue-Arbeiten wesentlich sicherer als ein vorschneller Header-Restore.

Detached Header

LUKS kann auch mit einem vom verschlüsselten Datenbereich getrennten Header verwendet werden.

In solchen Konfigurationen liegt der LUKS-Header nicht zwingend am Anfang des Datenblocks.

Die Option:

`--header`

ermöglicht die Angabe eines separaten Header-Geräts beziehungsweise einer Header-Datei.

Bedeutung für `isLuks`

Wird ein Detached Header verwendet, kann eine reine Untersuchung des Datenbereichs ohne den separaten Header zu dem Eindruck führen, es liege kein normal erkennbarer LUKS-Header vor.

Die tatsächliche Systemkonfiguration ist daher bei ungewöhnlichen Volumes zu berücksichtigen.

Keyfile

Neben interaktiv eingegebenen Passphrasen kann Cryptsetup Schlüsselmaterial auch aus Dateien beziehen.

Ein solches Keyfile ist sicherheitskritisch.

Wer Zugriff auf ein gültiges Keyfile erhält, kann – abhängig von der LUKS-Konfiguration – das Volume entsperren.

Keyfile nicht mit Header-Backup verwechseln

Ein:

Keyfile

und ein:

LUKS-Header-Backup

sind unterschiedliche Objekte.

Beide können jedoch sicherheitskritisch sein.

Schlüssel hinzufügen

Cryptsetup unterstützt:

`luksAddKey`

Damit kann einem bestehenden LUKS-Volume ein weiterer Schlüsselzugang hinzugefügt werden.

Diese Operation verändert die LUKS-Metadaten.

Schlüssel entfernen

Mit:

`luksRemoveKey`

kann ein vorhandener Schlüsselzugang entfernt werden.

Auch dies ist eine schreibende Verwaltungsoperation.

Keyslot gezielt löschen

Mit:

`luksKillSlot DEVICE NUMMER`

kann ein bestimmter Keyslot gelöscht werden.

Dies ist besonders gefährlich, wenn nicht sicher bekannt ist, welche weiteren funktionierenden Schlüsselzugänge existieren.

Passphrase ändern

`luksChangeKey`

ändert einen vorhandenen Schlüsselzugang.

Auch dies gehört nicht zu einer normalen read-only Rescue-Analyse.

Rescue-Grundsatz für Keyslots

Bei einer Datenrettung gilt:

Keine Keyslots verändern, solange das Ziel ausschließlich darin besteht, vorhandene Daten zu lesen und zu sichern.

Jede unnötige Veränderung erhöht das Risiko.

`luksFormat`

Die Operation:

`cryptsetup luksFormat DEVICE`

initialisiert ein Gerät als neues LUKS-Volume.

Dies ist eine **destruktive Erstellungsoperation**.

Niemals zur „Reparatur“ verwenden

`luksFormat` ist kein Reparaturbefehl für einen beschädigten vorhandenen LUKS-Header.

Wird ein vorhandener Datenträger neu mit `luksFormat` initialisiert, werden für die bisherige Verschlüsselung entscheidende Metadaten überschrieben.

Das kann die Wiederherstellung vorhandener Daten unmöglich machen.

Besonders gefährlicher Denkfehler

Ein beschädigtes LUKS-Volume darf nicht nach dem Muster:

„Header kaputt – einfach neu formatieren“

behandelt werden.

Bei Verschlüsselung kann gerade das Überschreiben des Headers die noch vorhandene letzte Möglichkeit zur Datenrettung zerstören.

`erase` / `luksErase`

Cryptsetup besitzt außerdem eine Operation zum Löschen der LUKS-Keyslots.

Sie macht ein LUKS-Volume ohne anderweitig verfügbaren Volume Key beziehungsweise geeignete Sicherungen unzugänglich.

Diese Funktion ist keine Rescue-Funktion.

Kein normales Datenlöschen

`luksErase` überschreibt nicht zwangsläufig den gesamten verschlüsselten Nutzdatenbereich.

Es zerstört den regulären Schlüsselzugang über die LUKS-Keyslots.

Vor der Verwendung muss das Sicherheitsmodell einschließlich vorhandener Header-Backups verstanden werden.

`convert`

Cryptsetup unterstützt die Konvertierung zwischen LUKS1 und LUKS2, soweit die Voraussetzungen erfüllt sind.

Eine Formatkonvertierung verändert kritische Metadaten.

Sie gehört deshalb nicht zu einer normalen Rescue-Erstanalyse.

`config`

Bei LUKS2 können mit:

```
cryptsetup config
```

permanente Konfigurationsoptionen im LUKS-Header verändert werden.

Auch dies ist eine schreibende Verwaltungsfunktion.

Tokens

LUKS2 kann Tokenobjekte verwenden, über die zusätzliche Mechanismen zur Beschaffung von Passphrasen beziehungsweise Schlüsselinformationen eingebunden werden können.

Cryptsetup besitzt dafür Token-Verwaltungsfunktionen.

Für eine Rescue-Analyse gilt wiederum:

Vorhandene Tokeninformationen untersuchen, aber nicht ohne Not verändern.

Plain dm-crypt

Cryptsetup unterstützt neben LUKS auch:

plain dm-crypt.

Plain dm-crypt besitzt im Gegensatz zu LUKS keine entsprechenden selbstbeschreibenden LUKS-Metadaten auf dem Datenträger.

Konsequenz bei Plain dm-crypt

Ohne die korrekten extern bekannten Parameter kann ein Plain-dm-crypt-Datenträger nicht einfach anhand eines LUKS-Headers erkannt und automatisch korrekt geöffnet werden.

Benötigt werden die zur ursprünglichen Einrichtung passenden Parameter.

Keine sichere automatische Erkennung von Verschlüsselung allein anhand „zufälliger Daten“

Verschlüsselte Daten sollen gerade wie hochentropische Daten erscheinen.

Aus einem Blockbereich, der scheinbar zufällige Bytes enthält, kann deshalb nicht zuverlässig allein geschlossen werden:

welcher Verschlüsselungsmodus,

welcher Schlüssel,

welche Parameter

oder überhaupt welches Verschlüsselungssystem

verwendet wurde.

LUKS erleichtert die Verwaltung

Genau hier liegt ein wesentlicher Vorteil von LUKS:

Die notwendigen Verwaltungsmetadaten werden in einem definierten Format gespeichert.

Die geheimen Zugangsdaten selbst müssen dennoch vorhanden sein.

Keine Passwortwiederherstellung

Cryptsetup ist kein Werkzeug zum „Zurücksetzen“ eines unbekannten LUKS-Passworts.

Ist kein gültiger Schlüsselzugang vorhanden, kann Cryptsetup die Verschlüsselung nicht einfach umgehen.

Das ist eine grundlegende Eigenschaft sicherer Verschlüsselung.

Verschlüsselung und Datenrettung

Bei einem verschlüsselten Datenträger existieren zwei voneinander unabhängige Schadensklassen:

Verschlüsselungsebene beschädigt

beispielsweise Header oder Keyslots.

und:

Datenebene beschädigt

beispielsweise Dateisystem oder Dateien innerhalb des entschlüsselten Volumes.

Beide Ebenen müssen separat diagnostiziert werden.

Beispiel 1: LUKS intakt, Dateisystem beschädigt

`isLuks` erfolgreich.

`luksDump` erfolgreich.

`open` erfolgreich.

Dateisystemprüfung zeigt Schäden.

Dann liegt das primäre Problem oberhalb der Verschlüsselungsschicht.

Beispiel 2: LUKS-Header beschädigt

`isLuks` schlägt unerwartet fehl.

Bekannt ist jedoch, dass der Datenträger zuvor LUKS verwendet hat.

Dann sollte keinesfalls sofort formatiert werden.

Stattdessen:

Original sichern,

Image erstellen,

vorhandene Header-Backups suchen,

Systemdokumentation prüfen

und erst danach weitere Wiederherstellungsmaßnahmen planen.

Beispiel 3: Passphrase unbekannt

LUKS wird erkannt.

Header ist lesbar.

Es ist jedoch kein gültiger Schlüsselzugang bekannt.

Dann ist die technische Verschlüsselungsstruktur vorhanden, aber Cryptsetup kann das Volume ohne gültige Authentisierung nicht regulär öffnen.

Das ist kein Dateisystemfehler.

Beispiel 4: Physisch defektes verschlüsseltes Laufwerk

SMART beziehungsweise Hardwarediagnose zeigt Probleme.

Lesevorgänge schlagen wiederholt fehl.

Dann:

nicht zuerst umfangreiche Cryptsetup- oder Dateisystemtests durchführen.

Stattdessen:

GNU ddrescue → Image/Klon → Cryptsetup auf der Kopie → Dateisystemanalyse.

Beispiel 5: Volume lässt sich öffnen

Passphrase funktioniert.

`/dev/mapper/rescue` erscheint.

Dann sollte zunächst geprüft werden, welche Struktur sich innerhalb dieses Mappings befindet.

Nicht automatisch formatieren oder Reparaturprogramme starten.

Cryptsetup und GNU ddrescue

GNU ddrescue arbeitet auf der Blockebene.

Cryptsetup arbeitet auf der Verschlüsselungsebene.

Für beschädigte verschlüsselte Datenträger ist deshalb die Reihenfolge häufig:

ddrescue → cryptsetup → Dateisystemwerkzeuge.

Cryptsetup und SMART/NVMe Diagnose

SMART beziehungsweise NVMe Diagnose helfen festzustellen, ob ein physischer Datenträger Anzeichen eines Hardwareproblems zeigt.

Cryptsetup kann dagegen feststellen, ob die Verschlüsselungsschicht zugänglich ist.

Beides beantwortet unterschiedliche Fragen.

Cryptsetup und GParted

GParted kann Partitionen und bestimmte Verschlüsselungsstrukturen erkennen beziehungsweise in seine Datenträgerdarstellung einbeziehen.

Cryptsetup ist jedoch das spezialisierte Kommandozeilenwerkzeug für die eigentliche dm-crypt-/LUKS-Verwaltung.

Cryptsetup und GNOME Disks

GNOME Disks kann verschlüsselte Volumes grafisch in den Datenträgerworkflow integrieren.

Cryptsetup bietet dagegen die präzise Kommandozeilenkontrolle und detaillierte LUKS-Verwaltungsfunktionen.

Cryptsetup und GnuPG

Cryptsetup und GnuPG gehören beide zum Bereich Verschlüsselung, lösen jedoch unterschiedliche Aufgaben.

Cryptsetup/LUKS

verschlüsselt Blockgeräte beziehungsweise Datenträger.

GnuPG

arbeitet unter anderem mit Dateien, Nachrichten, Signaturen und OpenPGP-Schlüsseln.

Ein LUKS-Volume ist daher nicht einfach eine „mit GPG verschlüsselte Festplatte“.

Typischer sicherer Rescue-Workflow

1. Datenträger identifizieren

Gerätebezeichnung, Größe und Hardware eindeutig prüfen.

2. Hardwarezustand beurteilen

Bei Auffälligkeiten zuerst Image erstellen.

3. Keine schreibenden Operationen durchführen

4. LUKS prüfen

```
cryptsetup -v isLuks DEVICE
```

5. Metadaten dokumentieren

```
cryptsetup luksDump DEVICE
```

6. LUKS UUID dokumentieren

```
cryptsetup luksUUID DEVICE
```

7. Wenn gültiger Schlüssel vorhanden: read-only öffnen

```
cryptsetup open --readonly --type luks DEVICE rescue
```

8. Mapping kontrollieren

```
cryptsetup status rescue
```

9. Innere Struktur untersuchen

10. Daten auf separates Zielmedium retten

11. Dateisystem ordnungsgemäß aushängen

12. Mapping schließen

```
cryptsetup close rescue
```

Grenzen

Cryptsetup repariert keinen physisch defekten Datenträger.

Cryptsetup ersetzt GNU ddrescue nicht.

Cryptsetup repariert kein beschädigtes Dateisystem.

Cryptsetup ersetzt Dateisystemwerkzeuge nicht.

Cryptsetup kann eine unbekannte starke Passphrase nicht einfach umgehen.

Ein erfolgreicher `luksDump` beweist nicht, dass die Nutzdaten vollständig intakt sind.

Ein erfolgreiches `open` beweist nicht, dass das Dateisystem intakt ist.

Ein negativer `isLuks`-Test beweist bei einem beschädigten Datenträger nicht sicher, dass niemals LUKS vorhanden war.

Plain dm-crypt besitzt keine mit LUKS vergleichbare selbstbeschreibende Headerstruktur.

Sicherheitshinweise

Bei Datenrettung grundsätzlich zunächst read-only denken.

Gerätebezeichnungen vor jedem Befehl doppelt prüfen.

`luksFormat` niemals als Reparaturversuch für ein vorhandenes verschlüsseltes Volume verwenden.

Keyslots nicht während einer normalen Datenrettung verändern.

`luksKillSlot`, `luksRemoveKey`, `luksErase`, `convert`, `config` und Header-Restore nur mit vollständigem Verständnis der Auswirkungen einsetzen.

Vor kritischen Änderungen Header-Backup und vollständige Datensicherung vorsehen.

Header-Backups wie sensibles Schlüsselmaterial schützen.

Passphrasen und Keyfiles niemals in Dokumentationen oder Logs aufnehmen.

Debugging-Werkzeuge können sensible Daten offenlegen; insbesondere externe Trace-Ausgaben sind mit großer Vorsicht zu behandeln.

Bei physisch beschädigten Medien zuerst ein Image beziehungsweise einen Klon erstellen.

Nicht auf dem Original experimentieren, wenn eine Kopie möglich ist.

Ein entsperrtes `/dev/mapper`-Gerät enthält die Daten aus Sicht nachfolgender Programme entschlüsselt und muss entsprechend geschützt werden.

Technische Einordnung in JUMBO BLOCK Rescue

Cryptsetup eröffnet im JUMBO BLOCK Rescue Handbuch den Bereich:

Verschlüsselung.

Seine zentrale Aufgabe ist die kontrollierte Überbrückung zwischen:

verschlüsseltem Blockgerät

und:

entschlüsselter Datenebene.

Die Rescue-Kette lautet bei einem verschlüsselten beschädigten Datenträger idealtypisch:

SMART / NVMe Diagnose

physische Situation beurteilen.

↓

GNU ddrescue

bei instabilem Medium zunächst eine sektorweise Arbeitskopie erzeugen.

↓

Cryptsetup `isLuks` / `luksDump`

Verschlüsselungsstruktur erkennen und dokumentieren.

↓

Cryptsetup `open --readonly`

Volume mit vorhandener gültiger Authentisierung kontrolliert öffnen.

↓

/dev/mapper/...

entschlüsselte Blockebene bereitstellen.

↓

Dateisystemanalyse

innere Struktur untersuchen.

↓

Datenrettung

Dateien auf ein separates Zielmedium sichern.

Damit nimmt Cryptsetup eine entscheidende Position im Rescue-System ein:

Ohne das Öffnen der Verschlüsselungsschicht sehen nachfolgende Dateisystem- und Datenrettungswerkzeuge lediglich die verschlüsselten Blöcke.

Die wichtigste Regel für den Rescue-Einsatz lautet deshalb:

Erkennen und lesen vor verändern.

Und bei beschädigten verschlüsselten Medien noch strenger:

Zuerst sichern – dann entschlüsseln – erst danach reparieren.

A.26 GnuPG

Bereich: Verschlüsselung
Programm: GnuPG – GNU Privacy Guard
Baseline 20261001: GnuPG 2.4.7
Installierte Kernpakete: `gnupg 2.4.7-21+deb13u1`, `gnupg2 2.4.7-21+deb13u1`, `gpg 2.4.7-21+deb13u1+b5`, `gpg-agent 2.4.7-21+deb13u1+b5`, `gpgv 2.4.7-21+deb13u1+b5`, `gpgsm 2.4.7-21+deb13u1+b5`, `gpgconf 2.4.7-21+deb13u1+b5`
Bedienung: Kommandozeile
Aufgabe: OpenPGP-Signaturprüfung, Datei-Ver- und Entschlüsselung sowie Schlüsselverwaltung

Zweck

GnuPG – GNU Privacy Guard – ist eine freie Implementierung kryptografischer Standards wie OpenPGP und S/MIME.

Für JUMBO BLOCK Rescue ist insbesondere das Programm:

`gpg`

relevant.

Es ermöglicht unter anderem:

digitale Signaturen zu prüfen,

Dateien zu signieren,

Dateien für Empfänger zu verschlüsseln,

verschlüsselte Dateien zu entschlüsseln,

symmetrisch mit einer Passphrase zu verschlüsseln,

öffentliche Schlüssel zu importieren und zu exportieren,

Fingerprints anzuzeigen

und Schlüsselbestände zu verwalten.

Zusätzlich steht mit:

`gpgv`

ein spezialisiertes Werkzeug ausschließlich zur OpenPGP-Signaturprüfung zur Verfügung.

Einsatzgebiet im Rescue-System

GnuPG eignet sich innerhalb von JUMBO BLOCK Rescue insbesondere für:

Prüfung digital signierter Downloads,

Verifikation von Softwarepaketen und Images,

Prüfung signierter Herstellerdateien,

Kontrolle von OpenPGP-Signaturen,

Verifikation einer Datei gegen eine separate `.sig`- oder `.asc`-Datei,

Identifikation des verwendeten Signaturschlüssels,

Vergleich von Schlüsselfingerprints,

Entschlüsselung autorisiert vorliegender OpenPGP-Dateien,

Erzeugung eigener Signaturen,

Verschlüsselung sensibler Rescue-Daten

und sichere Übertragung von Dateien an bekannte Empfänger.

Cryptsetup und GnuPG unterscheiden

Obwohl beide Programme im Bereich Verschlüsselung stehen, erfüllen sie grundlegend unterschiedliche Aufgaben.

Cryptsetup

arbeitet primär auf Blockgeräteebene.

Typischer Anwendungsfall:

verschlüsselte Festplatte oder Partition.

GnuPG

arbeitet primär mit Daten, Dateien, Nachrichten, Signaturen und kryptografischen Schlüsseln.

Typischer Anwendungsfall:

`rescue-image.iso`

plus:

`rescue-image.iso.sig`

Cryptsetup beantwortet damit beispielsweise:

Wie öffne ich dieses verschlüsselte Volume?

GnuPG beantwortet beispielsweise:

Ist diese Datei tatsächlich mit dem erwarteten OpenPGP-Schlüssel signiert worden?

Digitale Signatur

Eine digitale Signatur verbindet kryptografisch:

Daten

und:

einen privaten Signaturschlüssel.

Die Signatur kann anschließend mit dem zugehörigen öffentlichen Schlüssel überprüft werden.

Was eine erfolgreiche Signaturprüfung technisch aussagt

Eine erfolgreiche kryptografische Prüfung zeigt, dass:

die geprüften Daten zur Signatur passen

und:

die Signatur mit dem zum verwendeten öffentlichen Schlüssel gehörenden privaten Schlüssel erzeugt wurde.

Entscheidend bleibt jedoch die nächste Frage:

Ist dieser öffentliche Schlüssel tatsächlich der Schlüssel der erwarteten Person, Organisation oder des erwarteten Herstellers?

Signaturprüfung ist daher zweistufig

Eine belastbare Prüfung besteht aus zwei getrennten Fragen:

1. Ist die Signatur kryptografisch gültig?

und:

2. Ist der verwendete Schlüssel authentisch der erwarteten Quelle zugeordnet?

Nur die erste Frage allein reicht für eine sichere Herkunftsbewertung nicht aus.

Detached Signature

Bei Softwaredownloads wird häufig eine:

Detached Signature

verwendet.

Dabei liegen Nutzdatei und Signatur getrennt vor.

Beispiel:

`rescue.iso`

und:

`rescue.iso.sig`

Die Signaturdatei enthält nicht die komplette Nutzdatei, sondern die für die Signaturprüfung erforderlichen OpenPGP-Daten.

Detached Signature prüfen

Die offizielle GnuPG-Syntax lautet:

```
gpg --verify SIGNATUR DATEI
```

Beispielprinzip:

```
gpg --verify rescue.iso.sig rescue.iso
```

GnuPG empfiehlt bei Detached Signatures ausdrücklich, beide Dateinamen anzugeben.

Warum beide Dateien ausdrücklich angeben?

GnuPG besitzt historische Mechanismen, bei nur einer angegebenen Signaturdatei den Namen der zugehörigen Datendatei abzuleiten.

Die aktuelle GnuPG-Dokumentation rät bei Detached Signatures ausdrücklich davon ab, sich darauf zu verlassen.

Für einen reproduzierbaren Rescue-Workflow sollte deshalb immer angegeben werden:

Signaturdatei

und:

zu prüfende Datendatei.

Beispiel für eine ISO-Prüfung

Ein typischer Ablauf lautet:

```
gpg --verify image.iso.sig image.iso
```

GnuPG prüft dabei die Signatur gegen exakt die angegebene ISO-Datei.

Schon eine signaturrelevante Veränderung der Datei führt dazu, dass die Signatur nicht mehr korrekt verifiziert.

„Good signature“

Eine Meldung über eine gute Signatur bestätigt die kryptografische Übereinstimmung.

Sie darf jedoch nicht isoliert betrachtet werden.

Entscheidend ist zusätzlich:

Welcher Schlüssel hat signiert?

Fingerprint

Ein öffentlicher OpenPGP-Schlüssel besitzt einen Fingerprint.

Dieser Fingerprint dient als wesentlich präzisere Identifikation des Schlüssels als beispielsweise:

Name,

E-Mail-Adresse

oder eine kurze Key-ID.

Fingerprints können mit:

```
gpg --fingerprint
```

angezeigt werden. Die offizielle GnuPG-Dokumentation führt diese Funktion ausdrücklich als Standardverfahren zur Schlüsseldarstellung auf.

Warum der Fingerprint wichtig ist

Ein Schlüssel kann eine User ID enthalten wie:

```
Example Vendor <security@example.com>
```

Diese Zeichenfolge allein beweist nicht, dass der Schlüssel tatsächlich vom genannten Hersteller stammt.

Jeder könnte grundsätzlich einen eigenen OpenPGP-Schlüssel mit einer frei gewählten User ID erzeugen.

Deshalb muss die Schlüsselidentität über eine vertrauenswürdige Quelle überprüft werden.

Vertrauenswürdiger Vergleich

Der Fingerprint sollte möglichst über einen unabhängigen, authentisierten Kanal bezogen werden.

Beispiele können sein:

offizielle Herstellerdokumentation,

bereits vertrauenswürdig vorhandene Schlüssel,

offiziell veröffentlichte Releaseinformationen

oder ein anderer unabhängig geprüfter Kommunikationskanal.

Zentrale Rescue-Regel

Eine gültige Signatur mit einem unbekannten Schlüssel beweist nicht automatisch die erwartete Herkunft.

Sie beweist zunächst nur die kryptografische Beziehung zwischen:

Datei,

Signatur

und:

dem verwendeten Schlüssel.

Öffentliche Schlüssel anzeigen

Mit:

```
gpg --list-keys
```

werden öffentliche Schlüssel im aktuellen GnuPG-Schlüsselbestand angezeigt.

Ein bestimmter Schlüssel kann durch eine geeignete Suchangabe eingeschränkt werden.

Geheime Schlüssel anzeigen

Mit:

```
gpg --list-secret-keys
```

werden vorhandene geheime Schlüssel beziehungsweise geheime Schlüsselkomponenten aufgelistet.

Diese Information ist sicherheitskritischer als eine reine Liste öffentlicher Schlüssel.

Schlüssel importieren

Ein öffentlicher Schlüssel kann beispielsweise mit:

```
gpg --import DATEI
```

importiert werden.

Vor dem Vertrauen in diesen Schlüssel muss seine Authentizität separat geprüft werden.

Import ist kein Vertrauensbeweis

Ein erfolgreicher:

```
gpg --import vendor-key.asc
```

bedeutet lediglich:

GnuPG konnte die enthaltenen Schlüsseldaten in den Schlüsselbestand importieren.

Es bedeutet nicht:

Dieser Schlüssel gehört wirklich zum angegebenen Hersteller.

Schlüssel vor dem Import untersuchen

Bei fremden Schlüsseldateien ist es sinnvoll, deren Inhalt beziehungsweise Fingerprint zunächst zu untersuchen, bevor daraus Vertrauen abgeleitet wird.

Die Entscheidung über die Authentizität muss anhand einer unabhängigen Referenz erfolgen.

Öffentlichen Schlüssel exportieren

Öffentliche Schlüssel können exportiert werden.

Eine menschenlesbare ASCII-Armor-Darstellung kann beispielsweise mit:

```
--armor
```

erzeugt werden.

ASCII-armored OpenPGP-Daten sind typischerweise an Blöcken wie:

`-----BEGIN PGP PUBLIC KEY BLOCK-----`

erkennbar.

`--armor`

Mit:

`--armor`

beziehungsweise:

`-a`

erzeugt GnuPG ASCII-armored statt binärer OpenPGP-Ausgabe, soweit die betreffende Operation dies unterstützt.

Dies erleichtert beispielsweise die Übertragung kryptografischer Daten über textorientierte Kanäle.

Binär und ASCII Armor

Eine `.asc`-Datei ist häufig ASCII-armored.

Eine `.sig`-Datei kann eine binäre Detached Signature enthalten.

Die Dateiendung allein ist jedoch kein kryptografischer Beweis für den tatsächlichen Inhalt.

Datei signieren

Mit:

`gpg --sign DATEI`

kann eine Datei signiert werden.

Dabei entsteht eine OpenPGP-signierte Datenstruktur.

Detached Signature erzeugen

Für Softwaredateien ist häufig eine separate Signatur sinnvoll:

`gpg --detach-sign DATEI`

Kurzform:

`gpg -b DATEI`

Die offizielle GnuPG-Dokumentation beschreibt `--detach-sign` beziehungsweise `-b` ausdrücklich als Erzeugung einer Detached Signature.

ASCII-armored Detached Signature

In Verbindung mit:

`--armor`

kann eine textbasierte Detached Signature erzeugt werden.

Beispiel:

```
gpg --armor --detach-sign DATEI
```

Signaturschlüssel auswählen

Wenn mehrere geheime Schlüssel vorhanden sind, kann mit:

```
--local-user
```

beziehungsweise:

```
-u
```

der gewünschte Signaturschlüssel ausgewählt werden.

Dies verhindert, dass versehentlich ein anderer Standardschlüssel verwendet wird.

Cleartext Signature

Mit:

```
gpg --clear-sign DATEI
```

kann eine lesbare Textdatei mit einer OpenPGP-Cleartext-Signatur versehen werden.

Der Text bleibt dabei grundsätzlich direkt lesbar.

Cleartext Signatures und Binärdateien

Für Binärdateien wie:

ISO-Images,

Archive,

Programme

oder Firmware

ist eine Detached Signature typischerweise die klarere Struktur:

Originaldatei bleibt unverändert,

Signatur liegt separat vor.

Verifikation verändert die Datei nicht

```
gpg --verify
```

prüft eine Signatur, ohne die zu prüfende Datei umzuschreiben.

Damit eignet sich der Befehl besonders für Rescue- und Integritätsprüfungen.

GnuPG und Prüfsummen

Eine kryptografische Prüfsumme und eine digitale Signatur erfüllen unterschiedliche Aufgaben.

Ein veröffentlichter SHA-256-Wert kann beantworten:

Hat meine Datei denselben Hash wie der veröffentlichte Referenzwert?

Eine korrekt authentisierte digitale Signatur kann zusätzlich beantworten:

Wurde diese Datenfassung mit dem privaten Schlüssel der erwarteten Quelle signiert?

Prüfsumme ohne authentisierte Quelle

Wenn ein Angreifer sowohl die Download-Datei als auch den daneben veröffentlichten Hashwert manipulieren kann, schützt der bloße Vergleich beider Werte nicht vor dieser gemeinsamen Manipulation.

Eine digitale Signatur verwendet dagegen einen privaten Schlüssel, der nicht auf dem Downloadserver vorhanden sein muss.

Herstellerdownloads

Ein belastbarer Downloadworkflow kann deshalb lauten:

1. Datei aus der vorgesehenen Quelle beziehen

2. Signaturdatei beziehen

3. Hersteller-Schlüssel aus vertrauenswürdiger Quelle beziehen

4. Fingerprint unabhängig prüfen

5. Signatur prüfen

```
gpg --verify SIGNATUR DATEI
```

6. Ergebnis einschließlich Signaturschlüssel dokumentieren

Bereits vertrauenswürdiges GnuPG verwenden

Das GnuPG-Projekt weist bei seiner eigenen Integritätsprüfung auf einen wichtigen Bootstrapping-Grundsatz hin:

Eine neu heruntergeladene GnuPG-Version sollte nicht selbst als alleinige Grundlage dienen, um ihre eigene Authentizität zu bestätigen.

Zur Prüfung sollte eine bereits vertrauenswürdige GnuPG-Installation verwendet werden.

Bedeutung für JUMBO BLOCK Rescue

Eine fest in der geprüften Rescue-Baseline enthaltene GnuPG-Installation kann deshalb ein wertvolles Werkzeug sein, um später heruntergeladene signierte Dateien unabhängig vom Zielsystem zu überprüfen.

Voraussetzung bleibt, dass auch die verwendeten Signaturschlüssel authentisch geprüft wurden.

Datei für einen Empfänger verschlüsseln

Mit:

```
gpg --encrypt
```

beziehungsweise:

```
gpg -e
```

können Daten für einen oder mehrere öffentliche Schlüssel verschlüsselt werden.

Ein Empfänger wird typischerweise mit:

```
--recipient
```

beziehungsweise:

```
-r
```

angegeben.

Prinzip der Public-Key-Verschlüsselung

Zum Verschlüsseln wird der öffentliche Schlüssel des Empfängers verwendet.

Zum Entschlüsseln wird der passende geheime Schlüssel benötigt.

Dadurch muss der geheime Schlüssel nicht an den Absender weitergegeben werden.

Vor dem Verschlüsseln Empfängerschlüssel prüfen

Auch hier gilt:

Ein importierter öffentlicher Schlüssel mit passendem Namen ist nicht automatisch authentisch.

Wird für den falschen Schlüssel verschlüsselt, kann die Datei möglicherweise nur der Besitzer dieses falschen privaten Schlüssels entschlüsseln.

Der Fingerprint des Empfängerschlüssels muss deshalb vor sicherheitskritischer Nutzung geprüft werden.

Entschlüsseln

Mit:

```
gpg --decrypt DATEI
```

beziehungsweise:

```
gpg -d DATEI
```

werden OpenPGP-verschlüsselte Daten entschlüsselt.

Standardmäßig schreibt `--decrypt` die entschlüsselten Daten auf die Standardausgabe, sofern kein anderes Ausgabeverhalten angegeben wird.

Ausgabe in Datei

Mit:

```
--output DATEI
```

beziehungsweise:

```
-o DATEI
```

kann eine Ausgabedatei angegeben werden.

Beispielprinzip:

```
gpg --output daten.tar --decrypt daten.tar.gpg
```

Signierte verschlüsselte Datei

Ist eine entschlüsselte Datei zusätzlich signiert, prüft `gpg --decrypt` die enthaltene Signatur ebenfalls.

Das Ergebnis dieser Signaturprüfung muss wiederum hinsichtlich des verwendeten Schlüssels bewertet werden.

Symmetrische Verschlüsselung

GnuPG kann Daten auch ohne Empfänger-Public-Key symmetrisch mit einer Passphrase verschlüsseln:

```
gpg --symmetric DATEI
```

Kurzform:

```
gpg -c DATEI
```

Die Entschlüsselung erfordert dann die passende Passphrase.

Symmetrische Verschlüsselung im Rescue-Einsatz

Dies kann beispielsweise sinnvoll sein, wenn sensible Diagnose- oder Exportdateien auf einem Transportmedium geschützt werden müssen und kein geeignetes Public-Key-Verfahren vorgesehen ist.

Passphrase

Die Sicherheit symmetrischer Verschlüsselung hängt wesentlich von der Qualität und Geheimhaltung der Passphrase ab.

Eine kurze oder leicht erratbare Passphrase kann den Schutz erheblich schwächen.

Passphrase nicht in Kommandozeile schreiben

Sensible Passphrasen sollten nicht als Klartext in Shell-Kommandos, Skripten oder Dokumentationen hinterlegt werden.

Kommandozeilen können abhängig vom System in:

Shell-History,

Prozessinformationen,

Logs

oder Dokumentationen

auftauchen.

`gpg-agent`

Die Baseline enthält:

`gpg-agent`.

GnuPG verwendet den Agenten unter anderem für die Verwaltung geheimer Schlüsseloperationen und Passphraseninteraktion.

Passphrase-Caching

Der Agent kann Passphrasen zeitweise zwischenspeichern.

Das erhöht den Bedienkomfort, bedeutet aber:

Eine einmal eingegebene Passphrase muss während der Cache-Lebensdauer möglicherweise nicht bei jeder weiteren Operation erneut eingegeben werden.

Bedeutung im Rescue-System

Bei gemeinsam verwendeten oder unbeaufsichtigten Rescue-Systemen sollte berücksichtigt werden, dass eine zuvor freigeschaltete Schlüsseloperation gegebenenfalls noch über den Agenten verfügbar sein kann.

Agent beenden

GnuPG stellt über:

`gpgconf`

Mechanismen zur Verwaltung seiner Hintergrundkomponenten bereit.

Nach sensiblen Arbeiten kann es sinnvoll sein, vorhandene Agentzustände bewusst zu beenden beziehungsweise das Rescue-System kontrolliert herunterzufahren.

Private Schlüssel

Geheime OpenPGP-Schlüssel sind hochsensibles Material.

Wer einen privaten Signaturschlüssel kontrolliert, kann gültige Signaturen unter diesem Schlüssel erzeugen.

Wer einen privaten Entschlüsselungsschlüssel kontrolliert, kann für diesen Schlüssel verschlüsselte Daten entschlüsseln, soweit die übrigen Voraussetzungen erfüllt sind.

Private Schlüssel nicht unnötig importieren

Für die reine Prüfung einer Herstellersignatur wird nur der öffentliche Schlüssel benötigt.

Es gibt keinen Grund, dafür private Schlüssel auf das Rescue-System zu übertragen.

Public-Key-Verifikation ist read-only bezüglich des geprüften Artefakts

Eine Signaturprüfung benötigt keine Veränderung der zu prüfenden Datei.

Dies ist für Rescue-Arbeiten besonders wertvoll:

Ein Image, Archiv oder Firmwarepaket kann geprüft werden, ohne seinen Inhalt zu verändern.

gpgv

Die Baseline enthält außerdem:

`gpgv 2.4.7`.

`gpgv` ist eine reduzierte GnuPG-Komponente, die ausschließlich OpenPGP-Signaturen überprüft.

Unterschied gpg und gpgv

`gpg`

ist das umfassende OpenPGP-Werkzeug für:

Schlüsselverwaltung,

Signieren,

Verschlüsseln,

Entschlüsseln

und Verifizieren.

`gpgv`

ist auf:

Signaturverifikation

beschränkt.

Vertrauensmodell von gpgv

Ein wichtiger Unterschied:

gpgv behandelt die Schlüssel seines verwendeten Trusted-Keyrings als vertrauenswürdig.

Es verwendet damit ein bewusst vereinfachtes Vertrauensmodell und prüft laut GnuPG-Dokumentation insbesondere nicht selbst, ob diese Schlüssel abgelaufen oder widerrufen sind.

Konsequenz

Bei gpgv ist besonders wichtig, dass der verwendete Keyring bereits kontrolliert und vertrauenswürdig zusammengestellt wurde.

gpgv beantwortet nicht die vorgelagerte Frage:

Wie wurde die Echtheit dieses öffentlichen Schlüssels festgestellt?

gpgv mit Keyring

Mit:

`--keyring DATEI`

kann ein bestimmter vertrauenswürdiger Keyring für die Prüfung angegeben werden.

Mehrere `--keyring`-Optionen können kombiniert werden.

Detached Signature mit gpgv

Das Grundprinzip lautet:

`gpgv SIGNATUR DATEI`

Auch hier sollten Signatur und Datendatei bei Detached Signatures ausdrücklich angegeben werden.

Rückgabewert von gpgv

Die offizielle Dokumentation nennt:

`0` – Prüfung erfolgreich,

`1` – mindestens eine Signatur war ungültig,

andere Fehlercodes – schwerwiegende Fehler.

gpg und Exitcode

Bei gpg ist ein bloßer Prozess-Exitcode für automatisierte Signaturentscheidungen nicht ausreichend.

Die GnuPG-Dokumentation weist ausdrücklich darauf hin, dass bei Skripten bekannt sein muss:

was signiert wurde

und:

wer signiert hat.

Für automatisierte Verifikation sollen geeignete Statusinformationen oder das speziell dafür vorgesehene `gpgv` verwendet werden.

Signaturprüfung in Skripten

Ein Skript sollte deshalb nicht lediglich:

```
gpg --verify ...
```

ausführen und bei Exitcode 0 pauschal:

„Herstellerdatei authentisch“

melden.

Zusätzlich muss sichergestellt werden, dass der tatsächliche Signaturschlüssel der erwartete Schlüssel ist.

Schlüssel-ID versus Fingerprint

Kurze Key-IDs sind für eine belastbare Schlüsselidentifikation ungeeignet.

Für sicherheitsrelevante Prüfungen sollte mit dem vollständigen Fingerprint gearbeitet werden.

User IDs

Ein OpenPGP-Schlüssel kann eine oder mehrere User IDs enthalten.

Typischerweise enthalten diese:

Name,

E-Mail-Adresse

und optional weitere Angaben.

Diese Angaben sind Bestandteil der Schlüsselidentität innerhalb OpenPGP, aber nicht automatisch unabhängig verifiziert.

Vertrauen

GnuPG verwaltet neben der rein kryptografischen Schlüsselstruktur auch Vertrauensinformationen.

Dabei muss zwischen zwei Fragen unterschieden werden:

Ist eine Signatur mathematisch gültig?

und:

Wie stark vertraut GnuPG der Zuordnung dieses Schlüssels zu einer Identität?

Diese Aussagen sind nicht identisch.

„WARNING: This key is not certified with a trusted signature“

Eine solche Warnung bedeutet nicht automatisch:

Die Dateisignatur ist kryptografisch falsch.

Sie kann vielmehr bedeuten, dass GnuPG keine ausreichende Vertrauenskette für die Identitätszuordnung des Schlüssels kennt.

Deshalb müssen:

Signaturgültigkeit

und:

Schlüsselvertrauen

getrennt interpretiert werden.

Rescue-Workflow ohne aufgebautes Web of Trust

Für Herstellerdateien ist häufig ein pragmatischer und belastbarer Ansatz:

1. Hersteller-Public-Key beschaffen

2. vollständigen Fingerprint über einen unabhängigen vertrauenswürdigen Kanal verifizieren

3. Datei mit genau diesem Schlüssel prüfen

Damit hängt die Herkunftsbewertung nicht davon ab, ob auf dem Rescue-System bereits ein persönliches Web of Trust aufgebaut wurde.

Schlüsselserver

GnuPG kann über `dirmngr` öffentliche Schlüssel aus Netzwerkquellen beziehungsweise Schlüsselverzeichnisdiensten beziehen.

Die Baseline enthält `dirmngr` aus der GnuPG-2.4.7-Paketgeneration.

Schlüsselserver lösen das Authentizitätsproblem nicht

Dass ein Schlüssel von einem öffentlichen Schlüsselserver heruntergeladen werden kann, beweist nicht, dass er zur erwarteten Organisation gehört.

Auch nach einem Netzwerkabruf muss die Identität des Schlüssels anhand eines vertrauenswürdigen Fingerprints geprüft werden.

Netzwerk und Offline-Betrieb

Für die eigentliche Signaturprüfung benötigt GnuPG keine Internetverbindung, wenn:

zu prüfende Datei,

Signatur

und:

benötigter öffentlicher Schlüssel

bereits lokal vorliegen.

Dies ist für JUMBO BLOCK Rescue besonders wertvoll.

Offline-Verifikation

Ein vollständig offline durchführbarer Prüfablauf lautet:

DATEI

-

SIGNATUR

-

bereits verifizierter öffentlicher Schlüssel

→

GnuPG-Signaturprüfung

Damit muss das kompromittierungsverdächtige Zielsystem nicht für die Integritätsprüfung verwendet werden.

Beispiel: Rescue-ISO prüfen

Angenommen, es liegen vor:

`rescue.iso`

`rescue.iso.sig`

und ein zuvor authentisierter Hersteller-Public-Key.

Dann:

```
gpg --verify rescue.iso.sig rescue.iso
```

Anschließend wird geprüft:

Ist die Signatur gut?

Welcher Schlüssel wurde verwendet?

Stimmt dessen vollständiger Fingerprint mit der verifizierten Herstellerreferenz überein?

Beispiel: Signatur ist ungültig

Meldet GnuPG eine ungültige Signatur, darf die Datei nicht als unverändert gegenüber dem signierten Original behandelt werden.

Mögliche Ursachen sind:

Datei verändert,

falsche Signaturdatei,

Signatur beschädigt,

falsche Kombination von Datei und Signatur.

Beispiel: Public Key fehlt

Kann GnuPG die Signatur nicht prüfen, weil der öffentliche Schlüssel fehlt, ist dies nicht dasselbe wie:

Bad signature.

Die kryptografische Prüfung kann dann mangels Schlüssel nicht vollständig durchgeführt werden.

Der benötigte Schlüssel muss aus einer vertrauenswürdigen Quelle beschafft und authentisiert werden.

Beispiel: Good Signature, unbekannter Schlüssel

GnuPG meldet eine gute Signatur.

Der verwendete Schlüssel ist jedoch nicht als erwarteter Herstellerschlüssel verifiziert.

Dann lautet das Ergebnis nicht:

Herstellerdatei sicher bestätigt.

Sondern:

Signatur technisch gültig; Identität des Signaturschlüssels noch zu verifizieren.

Beispiel: Download mit Chromium

Ein typischer kombinierter Rescue-Workflow lautet:

Chromium

Datei und Signatur von der vorgesehenen Quelle herunterladen.

↓

GnuPG

Signatur prüfen.

↓

Fingerprint

gegen eine unabhängige Herstellerreferenz vergleichen.

↓

ClamAV

Datei bei Bedarf zusätzlich auf bekannte Schadsoftware untersuchen.

Diese Prüfungen beantworten unterschiedliche Fragen.

GnuPG und ClamAV

GnuPG

prüft kryptografische Signaturen beziehungsweise schützt Daten kryptografisch.

ClamAV

sucht nach bekannten beziehungsweise erkennbaren Schadsoftwaremerkmalen.

Eine gültig signierte Datei kann theoretisch trotzdem unerwünschtes Verhalten enthalten, wenn der legitime Herausgeber eine problematische Datei signiert hat oder sein Signaturschlüssel kompromittiert wurde.

Umgekehrt beweist ein fehlender Malwarefund nicht die Herkunft einer Datei.

GnuPG und Hashwerte

Hashvergleich

prüft Übereinstimmung mit einem Referenzhash.

Digitale Signatur

bindet die Daten kryptografisch an einen Signaturschlüssel.

Beides kann kombiniert werden.

GnuPG und Cryptsetup

Cryptsetup schützt:

Blockgeräte.

GnuPG schützt beziehungsweise signiert:

Daten und Dateien.

Ein sinnvoller Rescue-Fall kann deshalb beide Werkzeuge kombinieren:

Daten aus einem LUKS-Volume retten,

ausgewählte sensible Rescue-Dateien anschließend mit GnuPG verschlüsseln,

und diese kontrolliert an einen berechtigten Empfänger übertragen.

GnuPG und SSH

Der nachfolgende Remote-Bereich verwendet mit SSH ebenfalls Public-Key-Kryptografie.

OpenPGP-Schlüssel und normale SSH-Schlüssel sind jedoch unterschiedliche Schlüssel- und Protokollsysteme.

Sie dürfen nicht einfach gleichgesetzt werden.

Metadaten bleiben relevant

Verschlüsselung schützt nicht zwangsläufig sämtliche Metadaten außerhalb der verschlüsselten Nutzdaten.

Beispielsweise können:

Dateiname,

Dateigröße,

Zeitpunkte,

Speicherort

oder Übertragungsmetadaten

abhängig vom verwendeten Workflow weiterhin sichtbar sein.

Sichere Löschung

Das Löschen einer unverschlüsselten Originaldatei nach GnuPG-Verschlüsselung garantiert auf modernen Dateisystemen und insbesondere SSDs nicht automatisch, dass alle früheren Datenkopien physisch überschrieben wurden.

GnuPG ist kein Werkzeug für garantiertes physisches Secure Erase.

Rescue-System und Schlüsselmaterial

Werden eigene geheime Schlüssel auf JUMBO BLOCK Rescue importiert, muss bedacht werden, dass Schlüsselmaterial beziehungsweise Agentzustände auf dem verwendeten System verarbeitet werden.

Für reine Hersteller-Signaturprüfungen sollten deshalb ausschließlich öffentliche Schlüssel verwendet werden.

Grenzen

GnuPG repariert keine beschädigten Dateien.

GnuPG ist kein Malware-Scanner.

GnuPG ersetzt ClamAV nicht.

GnuPG ist kein Datenträgerverschlüsselungswerkzeug im Sinne von LUKS.

GnuPG ersetzt Cryptsetup nicht.

Eine gültige Signatur beweist nicht ohne zusätzliche Schlüsselprüfung die reale Identität des Herausgebers.

Ein importierter Public Key ist nicht automatisch vertrauenswürdig.

Ein Name oder eine E-Mail-Adresse in einer User ID ist kein Identitätsbeweis.

Ein Schlüsselserver ist keine automatische Vertrauensinstanz.

Verschlüsselung garantiert nicht die Löschung unverschlüsselter Vorversionen.

Verifikation kann nur die tatsächlich signierten Daten bewerten.

Sicherheitshinweise

Für sicherheitskritische Signaturprüfungen den vollständigen Fingerprint des erwarteten Schlüssels verifizieren.

Fingerprints möglichst über einen unabhängigen vertrauenswürdigen Kanal beziehen.

Bei Detached Signatures Signaturdatei und Nutzdatei ausdrücklich angeben.

Eine „Good signature“-Meldung niemals ohne Prüfung des Signaturschlüssels als vollständigen Herkunftsbeweis behandeln.

Private Schlüssel nur auf ein Rescue-System übertragen, wenn dies wirklich erforderlich ist.

Für reine Verifikation werden keine privaten Schlüssel benötigt.

Passphrasen nicht in Shell-History, Skripten oder Dokumentationen hinterlegen.

Schlüsseldateien und Backups sicher schützen.

Bei gemeinsam genutzten Rescue-Systemen mögliche Passphrase-Caches des `gpg-agent` berücksichtigen.

Heruntergeladene Schlüssel nicht allein aufgrund ihrer User ID als authentisch betrachten.

Unbekannte Schlüssel nicht blind als vertrauenswürdig markieren.

Signaturprüfung möglichst mit einer bereits vertrauenswürdigen GnuPG-Installation durchführen.

Technische Einordnung in JUMBO BLOCK Rescue

GnuPG schließt innerhalb von JUMBO BLOCK Rescue den Bereich:

Verschlüsselung.

Die beiden Programme dieses Bereichs besitzen klar getrennte Aufgaben:

A.25 Cryptsetup

Blockgeräteverschlüsselung und LUKS.

A.26 GnuPG

Dateiverschlüsselung, OpenPGP-Signaturen und Schlüsselprüfung.

Für JUMBO BLOCK Rescue ist insbesondere die Signaturprüfung von großer Bedeutung.

Ein typischer sicherer Downloadworkflow lautet:

Chromium

Datei herunterladen.

↓

GnuPG

digitale Signatur prüfen.

↓

Fingerprint

Signaturschlüssel unabhängig authentisieren.

↓

ClamAV

Datei gegebenenfalls zusätzlich auf bekannte Schadsoftware untersuchen.

↓

erst danach

Datei verwenden.

Für verschlüsselte Rescue-Daten lautet ein anderer Workflow:

Daten retten

↓

Empfängerschlüssel authentisieren

↓

mit GnuPG verschlüsseln

↓

verschlüsselte Datei übertragen

↓

berechtigter Empfänger entschlüsselt mit seinem privaten Schlüssel.

Die wichtigste Grundregel für GnuPG innerhalb von JUMBO BLOCK Rescue lautet:

Eine kryptografisch gültige Signatur ist nur so aussagekräftig wie die verifizierte Identität des Schlüssels, mit dem sie erzeugt wurde.

Damit liefert GnuPG nicht lediglich Verschlüsselung.

Es schafft bei korrekt verifiziertem Schlüssel eine kryptografisch prüfbare Verbindung zwischen:

Datei

und:

erwarteter Quelle.

A.27 SSH / OpenSSH

Bereich: Remote
Programm: SSH – OpenSSH Client
Baseline 20261001: OpenSSH 10.0p1
Installiertes Kernpaket: `openssh-client 1:10.0p1-7+deb13u4`
Zusätzlich belegt: `ssh-askpass-gnome 1:10.0p1-7+deb13u4`
Bedienung: Kommandozeile
Aufgabe: Verschlüsselter Remote-Zugang zu autorisierten SSH-Systemen, Remote-Kommandos, sichere Dateiübertragung und Tunnel

Zweck

SSH – Secure Shell – stellt eine verschlüsselte Verbindung zu einem entfernten System her.

Der in JUMBO BLOCK Rescue installierte OpenSSH-Client kann insbesondere verwendet werden für:

Remote-Shell-Zugriffe,

Ausführung einzelner Kommandos auf entfernten Systemen,

Authentisierung per Passwort oder Public Key, soweit der Server dies zulässt,

Übertragung von Dateien,

SFTP-Verbindungen,

Portweiterleitungen,

Zugriff über Jump Hosts

und Diagnose von SSH-Verbindungen.

Baseline-Abgrenzung

Die Baseline 20261001 belegt:

`openssh-client 1:10.0p1-7+deb13u4`

und:

`ssh-askpass-gnome 1:10.0p1-7+deb13u4`.

Ein installiertes Paket:

`openssh-server`

ist durch die vorliegende Paketbaseline nicht belegt.

Daraus folgt für dieses Kapitel:

JUMBO BLOCK Rescue wird hier als **SSH-Client-System** dokumentiert.

Eine standardmäßig auf JUMBO BLOCK Rescue laufende SSH-Serverfunktion wird nicht behauptet.

Grundsyntax

Eine einfache SSH-Verbindung wird aufgebaut mit:

```
ssh HOST
```

Soll ein bestimmter Benutzer verwendet werden:

```
ssh USER@HOST
```

Beispielprinzip:

```
ssh admin@server.example
```

Benutzername

Der Benutzername bezeichnet das Benutzerkonto auf dem entfernten System.

Er muss nicht mit dem lokalen Benutzernamen des Rescue-Systems identisch sein.

Alternativ kann der Remote-Benutzer über die SSH-Konfiguration festgelegt werden.

IP-Adresse statt Hostname

SSH kann auch direkt mit einer IP-Adresse verwendet werden:

```
ssh USER@IP-ADRESSE
```

Damit kann die DNS-Auflösung des Zielnamens für diesen Verbindungstest umgangen werden.

DNS und SSH

Wird ein Hostname verwendet, muss dieser zunächst in eine erreichbare Adresse aufgelöst werden.

Bei Problemen kann deshalb mit:

Dig

geprüft werden, ob die Namensauflösung korrekt funktioniert.

Standardport

SSH-Server verwenden üblicherweise TCP-Port:

22.

Ein Server kann jedoch auf einem anderen Port konfiguriert sein.

Anderen Port verwenden

Mit:

`-p PORT`

wird ein abweichender Zielport angegeben.

Beispiel:

```
ssh -p 2222 USER@HOST
```

SSH besteht aus mehreren Sicherheitsstufen

Beim Aufbau einer SSH-Verbindung müssen insbesondere zwei unterschiedliche Authentisierungsfragen getrennt werden:

1. Ist der Server wirklich der erwartete Server?

→ Host-Key-Prüfung.

2. Darf der Benutzer auf diesen Server zugreifen?

→ Benutzer-Authentisierung.

Diese beiden Vorgänge dürfen nicht miteinander verwechselt werden.

Serverauthentisierung

SSH-Server besitzen kryptografische Host Keys.

Der Client verwendet diese Schlüssel, um die Identität des Servers über wiederholte Verbindungen hinweg zu überprüfen.

Der Host Key authentisiert dabei den Server gegenüber dem Client.

Benutzer-Authentisierung

Anschließend muss sich der Benutzer gegenüber dem Server authentisieren.

Mögliche Verfahren hängen von der Serverkonfiguration ab.

Dazu gehören beispielsweise:

Public-Key-Authentisierung,

Passwortauthentisierung

und weitere vom Server unterstützte SSH-Authentisierungsverfahren.

Host Key und Benutzer-Key sind unterschiedliche Schlüssel

Ein häufiger Denkfehler ist, beide Schlüsselarten gleichzusetzen.

Host Key

identifiziert den SSH-Server.

User Key

authentisiert einen Benutzer gegenüber dem SSH-Server.

Beide lösen unterschiedliche Aufgaben.

Erste Verbindung

Bei der ersten Verbindung zu einem bisher unbekannten SSH-Host besitzt der Client normalerweise noch keinen gespeicherten Host Key für dieses Ziel.

Der Benutzer erhält dann abhängig von der Konfiguration Informationen zum neuen Host Key beziehungsweise dessen Fingerprint.

Fingerprint prüfen

Vor der erstmaligen Annahme eines sicherheitskritischen Host Keys sollte dessen Fingerprint über einen vertrauenswürdigen unabhängigen Kanal überprüft werden.

Beispielsweise über:

lokale Serverkonsole,

Dokumentation,

Configuration Management,

Inventarsystem

oder eine andere bereits authentisierte Informationsquelle.

Warum das wichtig ist

Wird bei der ersten Verbindung blind ein fremder Host Key akzeptiert, kann der Client nicht wissen, ob er tatsächlich mit dem erwarteten Server oder einer zwischengeschalteten Gegenstelle verbunden ist.

Verschlüsselung allein löst dieses Identitätsproblem nicht.

`known_hosts`

Bekannte SSH-Host Keys werden typischerweise im Benutzerkontext in:

`~/.ssh/known_hosts`

gespeichert.

OpenSSH kann außerdem globale Known-Hosts-Dateien verwenden.

Die Host-Key-Datenbank ermöglicht den Vergleich bei späteren Verbindungen.

Wiederholte Verbindung

Bei späteren Verbindungen kann SSH prüfen, ob der vom Server präsentierte Host Key zum gespeicherten Schlüssel passt.

Stimmt er überein, besitzt der Client eine Kontinuität zur zuvor akzeptierten Serveridentität.

Host Key hat sich geändert

Ändert sich ein zuvor gespeicherter Host Key, erzeugt OpenSSH eine deutliche Warnung.

Eine solche Warnung darf nicht routinemäßig beseitigt werden.

Legitime Ursachen einer Host-Key-Änderung

Eine Änderung kann beispielsweise auftreten nach:

Neuinstallation des Servers,

Austausch des Systems,

bewusster Regeneration der Host Keys,

Wiederherstellung aus einer anderen Konfiguration

oder Änderung der SSH-Infrastruktur.

Sicherheitsrelevante Ursache

Ein unerwartet geänderter Host Key kann jedoch auch auf eine falsche Gegenstelle oder einen Man-in-the-Middle-Angriff hinweisen.

Deshalb gilt:

Erst Ursache klären, dann Known-Hosts-Eintrag ändern.

Gespeicherten Host suchen

Mit:

```
ssh-keygen -F HOST
```

kann nach einem Host in einer Known-Hosts-Datei gesucht werden.

Diese Funktion funktioniert auch bei entsprechend gehashten Hostnamen.

Alten Host-Key-Eintrag entfernen

Mit:

```
ssh-keygen -R HOST
```

können gespeicherte Schlüssel für einen Host aus `known_hosts` entfernt werden.

Sicherheitsregel für `ssh-keygen -R`

Der Befehl sollte nicht einfach ausgeführt werden, um eine Host-Key-Warnung „wegzubekommen“.

Zuerst muss geklärt werden, warum sich der Host Key geändert hat.

Erst wenn die neue Serveridentität unabhängig bestätigt wurde, ist das Entfernen des veralteten Eintrags ein sinnvoller administrativer Schritt.

`StrictHostKeyChecking`

Das Verhalten bei unbekannten beziehungsweise geänderten Host Keys wird unter anderem über:

`StrictHostKeyChecking`

gesteuert.

Diese Einstellung ist ein zentraler Bestandteil der SSH-Serveridentitätsprüfung.

Host-Key-Prüfung nicht deaktivieren

Optionen, die Host-Key-Prüfungen pauschal umgehen, sollten nicht Bestandteil eines normalen Rescue-Workflows sein.

Gerade ein Rescue-System kann in unbekannten oder kompromittierten Netzwerken eingesetzt werden.

Die Serverauthentisierung ist dort besonders wichtig.

Public-Key-Authentisierung

Bei der Public-Key-Authentisierung besitzt der Benutzer ein kryptografisches Schlüsselpaar:

privater Schlüssel

und:

öffentlicher Schlüssel.

Der Server kennt beziehungsweise autorisiert den öffentlichen Schlüssel.

Der Client beweist den Besitz des zugehörigen privaten Schlüssels.

Privater Schlüssel

Der private Schlüssel ist geheim.

Er darf nicht:

unkontrolliert kopiert,

öffentlich gespeichert,

in Tickets eingefügt,

per ungeschützter Nachricht versendet

oder auf unbekannten Systemen abgelegt

werden.

Öffentlicher Schlüssel

Der öffentliche Schlüssel darf an Systeme verteilt werden, auf denen der Benutzer damit autorisiert werden soll.

Er ermöglicht nicht unmittelbar die Rekonstruktion des privaten Schlüssels.

Identity-Datei auswählen

Mit:

`-i`

kann eine bestimmte Identity-Datei für die Public-Key-Authentisierung ausgewählt werden.

Beispiel:

```
ssh -i ~/.ssh/id_ed25519 USER@HOST
```

OpenSSH unterstützt mehrere Identity-Dateien und kann diese auch hostbezogen konfigurieren.

Passphrase für private Schlüssel

Private SSH-Schlüssel können mit einer Passphrase geschützt werden.

Dadurch ist der Besitz der Schlüsseldatei allein nicht automatisch ausreichend, um sie zu verwenden.

Schlüssel ohne Passphrase

Ein privater Schlüssel ohne Passphrase kann automatisierte Nutzung vereinfachen.

Wird die Datei jedoch kopiert oder gestohlen, fehlt eine zusätzliche Schutzschicht.

Auf einem Rescue-System sollte privates Schlüsselmaterial deshalb besonders restriktiv behandelt werden.

`ssh-agent`

OpenSSH kann private Schlüssel über:

`ssh-agent`

bereitstellen.

Der Agent hält Identitäten für SSH-Authentisierung verfügbar.

`ssh-add`

Mit:

`ssh-add`

können Schlüssel einem laufenden Agenten hinzugefügt werden.

Mit:

`ssh-add -l`

werden die Fingerprints der aktuell im Agenten vorhandenen Identitäten angezeigt.

Agentzustand beachten

Wurde ein Schlüssel in den Agenten geladen, kann er für weitere SSH-Verbindungen verfügbar bleiben.

Auf einem Rescue-System ist daher nach sicherheitskritischen Arbeiten zu berücksichtigen, welche Identitäten noch im Agenten geladen sind.

Agent Forwarding

Mit:

```
ssh -A
```

kann die Verbindung zum lokalen Authentisierungsagenten an das entfernte System weitergereicht werden.

Dadurch muss der private Schlüssel selbst nicht auf dem Remote-System gespeichert werden.

Agent Forwarding ist sicherheitskritisch

OpenSSH warnt ausdrücklich vor unbedachter Verwendung von Agent Forwarding.

Ein Benutzer mit ausreichenden Rechten auf dem entfernten System kann auf den weitergereichten Agent-Socket zugreifen und den Agenten für Authentisierungsoperationen missbrauchen.

Der Angreifer erhält dadurch zwar nicht unmittelbar das private Schlüsselmaterial, kann aber Signatur-/Authentisierungsoperationen mit den geladenen Identitäten auslösen.

Rescue-Regel

Bei einem möglicherweise kompromittierten Zielsystem:

kein Agent Forwarding verwenden.

Jump Host statt Agent Forwarding

Für den Zugriff auf ein internes Ziel über ein vorgeschaltetes SSH-System kann:

```
-J
```

verwendet werden.

Beispielprinzip:

```
ssh -J USER@JUMPHOST USER@ZIEL
```

OpenSSH verbindet sich dabei zunächst mit dem Jump Host und richtet darüber eine TCP-Weiterleitung zum endgültigen Ziel ein.

Vorteil

Für diesen Anwendungsfall muss der lokale Authentisierungsagent nicht zwingend an den Jump Host weitergereicht werden.

Die OpenSSH-Dokumentation nennt Jump Hosts ausdrücklich als sicherere Alternative in Situationen, in denen sonst Agent Forwarding verwendet würde.

Remote-Shell

Der Standardaufruf:

```
ssh USER@HOST
```

öffnet nach erfolgreicher Verbindung und Authentisierung normalerweise eine interaktive Remote-Sitzung.

Die dort eingegebenen Kommandos laufen auf dem entfernten System.

Sehr wichtige Rescue-Regel

Nach dem Login muss jederzeit klar sein:

Arbeite ich gerade lokal auf JUMBO BLOCK Rescue oder remote auf dem Zielsystem?

Ein versehentlich auf dem falschen System ausgeführter Befehl kann erhebliche Schäden verursachen.

Remote-Kommando

SSH kann auch direkt ein einzelnes Kommando auf dem entfernten System ausführen.

Grundprinzip:

```
ssh USER@HOST COMMAND
```

In diesem Fall wird das angegebene Kommando remote ausgeführt, anstatt eine normale Login-Shell zu starten.

Beispielprinzip

```
ssh admin@server.example uname -a
```

Das Kommando:

```
uname -a
```

läuft auf dem entfernten Server.

Nicht auf dem Rescue-System.

Vorteil

Dies eignet sich beispielsweise für gezielte Diagnoseabfragen, ohne eine längere interaktive Remote-Shell zu öffnen.

Vorsicht mit Remote-Kommandos

Ein Remote-Kommando besitzt die Rechte des authentisierten Remote-Benutzers.

Mit `sudo` oder einem privilegierten Konto können entsprechend weitreichende Änderungen ausgelöst werden.

Pseudo-Terminal

Mit:

`-t`

kann SSH die Zuweisung eines Pseudo-Terminals erzwingen.

Dies ist beispielsweise für bestimmte interaktive Remote-Kommandos erforderlich.

Mit:

`-T`

wird die Pseudo-Terminal-Zuweisung deaktiviert.

IPv4 erzwingen

Mit:

`-4`

kann der SSH-Client auf IPv4 beschränkt werden.

Dies kann bei Diagnosefällen hilfreich sein, in denen IPv4 und IPv6 getrennt untersucht werden sollen.

IPv6 erzwingen

Mit:

`-6`

wird entsprechend IPv6 verwendet.

Verbose-Modus

Für die SSH-Diagnose ist:

`-v`

besonders wichtig.

Beispiel:

```
ssh -v USER@HOST
```

Mehrere `-v` erhöhen die Ausführlichkeit bis maximal drei Stufen:

`-vv`

`-vvv`.

Was Verbose-Ausgabe hilft zu unterscheiden

Die Debugausgabe kann unter anderem helfen festzustellen:

welche Konfiguration verwendet wird,

welche Adresse kontaktiert wird,

wie weit der Verbindungsaufbau kommt,

welche Host-Key- beziehungsweise Authentisierungsschritte stattfinden

und warum eine Anmeldung möglicherweise scheitert.

Debugausgabe kann sensible Informationen enthalten

Verbose-Logs sollten nicht unkontrolliert veröffentlicht werden.

Sie können Informationen enthalten über:

Benutzernamen,

Hostnamen,

interne Adressen,

Schlüsseldateinamen,

Authentisierungsmethoden

und Infrastrukturdetails.

Effektive SSH-Konfiguration anzeigen

Mit:

```
ssh -G HOST
```

gibt OpenSSH die nach Auswertung der Konfiguration resultierenden Clientparameter aus und beendet sich anschließend.

Warum `ssh -G` hilfreich ist

SSH-Konfiguration kann aus mehreren Quellen stammen.

Bei einem unerwarteten Verhalten lässt sich damit beispielsweise prüfen:

welcher Benutzer,

welcher Port,

welche Identity-Dateien,

welche Proxy-/Jump-Einstellungen

oder andere Parameter effektiv verwendet werden.

Benutzerkonfiguration

Die benutzerspezifische SSH-Konfiguration liegt typischerweise unter:

`~/.ssh/config`

Dort können hostbezogene Einstellungen definiert werden.

Beispielprinzip

Eine Konfiguration kann sinngemäß enthalten:

```
Host rescue-server
HostName 192.0.2.10
User admin
Port 22
IdentityFile ~/.ssh/id_ed25519
```

Anschließend genügt:

`ssh rescue-server`

Vorteil

Wiederkehrende Verbindungen müssen nicht jedes Mal mit sämtlichen Parametern auf der Kommandozeile angegeben werden.

Systemweite Konfiguration

OpenSSH berücksichtigt zusätzlich systemweite Clientkonfiguration.

Für eine genaue Diagnose sollte deshalb nicht angenommen werden, dass ausschließlich `~/.ssh/config` das Verhalten bestimmt.

`ssh -G HOST` ist gerade hierfür hilfreich.

Dateiübertragung mit `scp`

Das OpenSSH-Clientpaket stellt auch:

`scp`

zur Verfügung.

Damit können Dateien über eine SSH-Verbindung übertragen werden.

Moderne `scp`-Übertragung

Seit OpenSSH 9.0 verwendet `scp` standardmäßig das SFTP-Protokoll für Dateiübertragungen.

Damit gilt dies auch für die hier dokumentierte OpenSSH-10.0p1-Generation.

Lokale Datei auf Remote-System kopieren

Grundprinzip:

```
scp DATEI USER@HOST:/ZIELPFAD/
```

Remote-Datei lokal sichern

Grundprinzip:

```
scp USER@HOST:/PFAD/DATEI ./
```

Verzeichnisse

Mit:

```
scp -r
```

können Verzeichnisbäume rekursiv übertragen werden.

`scp` verwendet SSH-Sicherheit

`scp` verwendet die SSH-Verbindung und damit dieselbe grundlegende Authentisierung und Transportsicherheit wie eine SSH-Sitzung.

Port bei `scp`

Bei `scp` wird ein abweichender SSH-Port mit:

```
-P
```

angegeben.

Dies unterscheidet sich von:

`ssh -p`.

Groß-/Kleinschreibung ist hier relevant.

`sftp`

OpenSSH stellt außerdem den interaktiven SFTP-Client:

```
sftp
```

bereit.

SFTP ist ein Dateiübertragungsprotokoll über SSH.

SFTP ist nicht FTP über TLS

SFTP darf nicht mit:

FTP,

FTPS

oder einem einfachen „verschlüsselten FTP“

verwechselt werden.

SFTP ist ein eigenes Protokoll, das über SSH transportiert wird.

Rescue-Einsatz für Dateiübertragung

SSH-basierte Dateiübertragung eignet sich beispielsweise für:

Logs vom Zielsystem holen,

Konfigurationsdateien sichern,

Diagnoseergebnisse übertragen,

kleine Reparaturdateien bereitstellen

oder Daten auf ein autorisiertes Remote-System kopieren.

Vor dem Upload auf ein beschädigtes System

Bei einem Rescue-Fall sollte zunächst geklärt werden, ob auf das Zielsystem geschrieben werden darf.

Ein Upload verändert den Zielzustand.

Für:

forensische Untersuchungen,

Beweissicherung

oder besonders sensible Incident-Response-Fälle

kann dies unerwünscht sein.

Port Forwarding

SSH kann TCP-Verbindungen durch den verschlüsselten SSH-Kanal weiterleiten.

OpenSSH unterstützt unter anderem:

lokale,

entfernte

und dynamische Portweiterleitung.

Local Forwarding

Mit:

-L

wird ein lokaler Port über die SSH-Verbindung zu einem Ziel auf der entfernten Seite weitergeleitet.

Grundprinzip:

```
ssh -L LOKALER_PORT:ZIELHOST:ZIELPORT USER@SSH-HOST
```

Beispielanwendung

Ein internes Webinterface ist nur vom SSH-Server aus erreichbar.

Ein lokaler Tunnel kann dieses Interface kontrolliert über die SSH-Verbindung erreichbar machen.

Danach kann beispielsweise Chromium lokal auf den weitergeleiteten Port zugreifen.

Remote Forwarding

Mit:

```
-R
```

wird auf der entfernten Seite ein Port beziehungsweise Socket bereitgestellt, dessen Verbindungen durch den SSH-Kanal zur lokalen Seite weitergeleitet werden.

Sicherheitsrisiko bei Forwardings

Ein Forwarding kann Netzwerkgrenzen überbrücken.

Dadurch können Dienste erreichbar werden, die ohne den Tunnel nicht erreichbar wären.

Portweiterleitungen dürfen deshalb nur:

autorisiert,

bewusst,

mit kontrollierten Bind-Adressen

und für den notwendigen Zeitraum

eingerichtet werden.

Local Forwarding bindet standardmäßig lokal

OpenSSH bindet lokale Portweiterleitungen standardmäßig an die Loopback-Adresse, sofern die Konfiguration dies nicht verändert.

Das reduziert unbeabsichtigte Erreichbarkeit durch andere Rechner.

Dynamisches Forwarding

SSH unterstützt außerdem dynamische Portweiterleitung.

Dabei kann der SSH-Client als lokaler SOCKS-Proxy für Verbindungen über den SSH-Kanal dienen.

Auch dies kann Netzwerkgrenzen verändern und gehört nur in einen bewusst geplanten Rescue-Workflow.

`-N`

Bei reinen Forwarding-Verbindungen kann SSH so verwendet werden, dass kein Remote-Kommando ausgeführt wird.

Dies ist beispielsweise für einen ausschließlich als Tunnel verwendeten SSH-Kanal sinnvoll.

SSH-Tunnel ist kein VPN-Ersatz für jede Situation

SSH-Portweiterleitungen transportieren gezielt bestimmte Verbindungen.

Sie ersetzen nicht automatisch eine vollständige VPN-Konfiguration.

SSH und Nmap

Nmap kann prüfen, ob ein SSH-Dienst am autorisierten Ziel sichtbar ist.

Beispielsweise kann ein offener TCP-Port 22 auf einen möglichen SSH-Dienst hinweisen.

SSH übernimmt anschließend den eigentlichen authentisierten Verbindungsaufbau.

SSH und Dig

Dig kann prüfen, ob der SSH-Hostname korrekt aufgelöst wird.

Wenn:

```
ssh server.example
```

fehlschlägt, aber:

```
ssh IP-ADRESSE
```

funktioniert, ist die Namensauflösung eine wichtige Untersuchungsrichtung.

SSH und MTR

MTR kann bei entfernten Zielen helfen, Netzwerkpfad und Latenz zu untersuchen.

Ein SSH-Timeout muss nicht durch SSH selbst verursacht werden.

Das Problem kann bereits auf der Netzwerkebene liegen.

SSH und GUFW

Eine lokale Firewall kann ausgehende oder eingehende Netzwerkkommunikation beeinflussen.

Bei unerwarteten SSH-Verbindungsproblemen sollte deshalb auch die Firewallkonfiguration berücksichtigt werden.

SSH und TCPDump/Wireshark

TCPDump und Wireshark können den SSH-Verbindungsaufbau auf Paketebene untersuchen.

Die Nutzdaten einer korrekt verschlüsselten SSH-Sitzung sind dabei nicht einfach als Klartext sichtbar.

Dies ist gerade ein Zweck des Protokolls.

SSH und Chromium

Über Local Port Forwarding kann ein Webdienst hinter einem SSH-System lokal verfügbar gemacht und anschließend mit Chromium untersucht werden.

Damit können beide Programme in einem Rescue-Workflow kombiniert werden.

SSH und SSHFS

Das folgende Kapitel A.28 behandelt:

SSHFS.

SSHFS verwendet SSH/SFTP, um ein entferntes Dateisystem lokal einzubinden.

Die SSH-Sicherheitsgrundlagen dieses Kapitels gelten deshalb auch für SSHFS:

Host-Key-Prüfung,

Benutzerauthentisierung,

Schlüsselmanagement

und Netzwerkerreichbarkeit.

Möglicherweise kompromittiertes Zielsystem

Remote-Rescue-Arbeiten haben eine besondere Sicherheitsdimension.

Ist das Zielsystem möglicherweise kompromittiert, darf ihm nicht automatisch vertraut werden.

Keine privaten Schlüssel auf das Ziel kopieren

Ein privater SSH-Schlüssel des Rescue-Systems sollte nicht auf einen möglicherweise kompromittierten Remote-Host kopiert werden.

Sonst könnte ein Angreifer diesen Schlüssel übernehmen.

Kein Agent Forwarding zum kompromittierten Ziel

Ebenso sollte Agent Forwarding nicht zu einem nicht vertrauenswürdigen System aktiviert werden.

Der private Schlüssel selbst wird zwar nicht übertragen, aber der Remote-Host kann unter den von OpenSSH beschriebenen Bedingungen den weitergeleiteten Agenten für Authentisierungsoperationen verwenden.

Remote-Ausgaben sind nicht automatisch vertrauenswürdig

Wenn ein kompromittiertes System remote Befehle ausführt, stammen die Ergebnisse aus diesem möglicherweise manipulierten System.

Ein Angreifer mit ausreichender Kontrolle kann beispielsweise:

Programme verändern,

Ausgaben manipulieren,

Logs verändern

oder Prozesse verstecken.

SSH sichert die Verbindung zum Host.

SSH garantiert nicht die Integrität des Betriebssystems, das auf diesem Host läuft.

Wichtige Unterscheidung

SSH-Transport sicher

bedeutet nicht:

Remote-System vertrauenswürdig.

Typischer Rescue-Ablauf: Server erreichbar, SSH funktioniert nicht

1. DNS prüfen

```
dig HOST
```

2. Netzwerkpfad prüfen

MTR beziehungsweise geeignete Netzdiagnose.

3. Port prüfen

Nmap am autorisierten Ziel.

4. SSH ausführlich starten

```
ssh -vvv USER@HOST
```

5. Fehlerstufe identifizieren

Namensauflösung?

TCP-Verbindung?

Host Key?

Schlüsselaustausch?

Benutzerauthentisierung?

6. Konfiguration prüfen

```
ssh -G HOST
```

Typischer Rescue-Ablauf: neuer Server

1. Zieladresse eindeutig feststellen
2. SSH-Verbindung starten
3. angezeigten Host-Key-Fingerprint nicht blind akzeptieren
4. Fingerprint über unabhängige vertrauenswürdige Quelle prüfen
5. erst danach Host Key übernehmen
6. Benutzer authentisieren

Typischer Rescue-Ablauf: Host-Key-Warnung

1. Verbindung nicht einfach erzwingen
2. prüfen, ob Server neu installiert oder ersetzt wurde
3. aktuellen Host-Key-Fingerprint über unabhängigen Kanal beschaffen
4. alten und neuen Schlüssel vergleichen
5. nur bei bestätigter Änderung alten Known-Hosts-Eintrag entfernen

```
ssh-keygen -R HOST
```

6. neue Verbindung aufbauen und neuen Fingerprint prüfen

Typischer Rescue-Ablauf: Datei vom Server sichern

1. Serveridentität prüfen
2. mit autorisiertem Benutzer verbinden
3. Quelldatei möglichst nicht verändern
4. Datei mit `scp` oder `sftp` auf ein separates Rescue-Ziel kopieren
5. bei kritischen Dateien anschließend Hash beziehungsweise Signatur dokumentieren

Typischer Rescue-Ablauf: internes Webinterface

1. SSH-Zugriff auf autorisierten Jump-/Administrationshost herstellen
2. lokalen Portforward einrichten
3. nur benötigten lokalen Port binden
4. Chromium gegen den lokalen Forward verwenden
5. Tunnel nach Abschluss schließen

Typischer Rescue-Ablauf: kompromittierungsverdächtiger Server

1. Zielidentität prüfen
2. dedizierte Rescue-Zugangsdaten verwenden

3. kein Agent Forwarding

4. keine unnötigen privaten Schlüssel übertragen

5. Remote-Kommandos auf das erforderliche Minimum beschränken

6. Ergebnisse nicht ungeprüft als vertrauenswürdige Systemwahrheit behandeln

7. relevante Daten auf separates Analysemedium sichern

Grenzen

SSH repariert kein Netzwerk.

SSH ersetzt Nmap, MTR oder Dig nicht.

SSH garantiert nicht, dass das entfernte Betriebssystem vertrauenswürdig ist.

Eine verschlüsselte SSH-Verbindung schützt nicht vor einem bereits kompromittierten Remote-System.

Eine erfolgreiche Benutzeranmeldung beweist nicht allein, dass der Server der erwartete Server ist – dafür ist die Host-Key-Prüfung zuständig.

Eine erfolgreiche Host-Key-Prüfung beweist nicht, dass der angemeldete Benutzer autorisiert handeln darf – dafür ist die Benutzer-Authentisierung beziehungsweise Berechtigungsstruktur zuständig.

SSH-Portweiterleitungen können Netzwerkgrenzen verändern.

Agent Forwarding kann zusätzliche Sicherheitsrisiken erzeugen.

Sicherheitshinweise

Host-Key-Warnungen niemals routinemäßig ignorieren.

Fingerprints neuer sicherheitskritischer Hosts unabhängig verifizieren.

`known_hosts` nicht pauschal löschen, um Verbindungsprobleme zu beseitigen.

Private SSH-Schlüssel vertraulich behandeln.

Private Schlüssel möglichst mit Passphrase schützen.

Keine privaten Schlüssel auf kompromittierungsverdächtige Zielsysteme kopieren.

Agent Forwarding nur einsetzen, wenn es wirklich erforderlich und das Ziel vertrauenswürdig ist.

Bei kompromittierungsverdächtigen Systemen Agent Forwarding vermeiden.

Remote-Kommandos vor Ausführung sorgfältig prüfen.

Immer unterscheiden, ob ein Kommando lokal oder remote ausgeführt wird.

Portforwardings nur autorisiert und mit minimal notwendiger Reichweite einrichten.

Verbose-Logs vor Weitergabe auf sensible Infrastrukturinformationen prüfen.

Technische Einordnung in JUMBO BLOCK Rescue

SSH eröffnet im JUMBO BLOCK Rescue Handbuch den Bereich:

Remote.

Die Baseline belegt dafür den:

OpenSSH Client 10.0p1.

Sie belegt dagegen in der vorliegenden Paketliste keinen OpenSSH-Server.

Die technische Rolle von A.27 lautet deshalb:

JUMBO BLOCK Rescue → sichere ausgehende Remote-Verbindung → autorisiertes Zielsystem.

Ein typischer Diagnosepfad lautet:

Dig

Zielnamen prüfen.

↓

MTR

Netzwerkpfad untersuchen.

↓

Nmap

SSH-Port beziehungsweise Dienst prüfen.

↓

SSH

Serveridentität über Host Key prüfen.

↓

Benutzer authentisieren

Passwort, Public Key oder anderes zugelassenes Verfahren.

↓

Remote-Diagnose

Shell oder gezielte Kommandos.

↓

SCP/SFTP

benötigte Dateien kontrolliert übertragen.

↓

SSHFS

bei Bedarf im nächsten Kapitel ein entferntes Dateisystem lokal einbinden.

Die zentrale Sicherheitslogik von SSH besteht dabei aus zwei unabhängigen Vertrauensrichtungen:

Host Key

Der Client prüft:

Bin ich mit dem erwarteten Server verbunden?

und:

Benutzer-Authentisierung

Der Server prüft:

Darf dieser Benutzer auf mich zugreifen?

Für den Rescue-Einsatz gilt deshalb als wichtigste Regel:

Eine verschlüsselte Verbindung ist erst dann wirklich hilfreich, wenn auch die Identität der Gegenstelle geprüft wurde.

A.28 SSHFS

Bereich: Remote
Programm: SSHFS
Baseline 20261001: SSHFS 3.7.3
Installiertes Paket: `sshfs 3.7.3-1.2~deb13u1`
FUSE-Basis der Baseline: `fuse3 3.17.2-3`, `libfuse3-4 3.17.2-3`
Bedienung: Kommandozeile
Aufgabe: Entfernte Verzeichnisse über SSH/SFTP als lokales Dateisystem einbinden

Zweck

SSHFS ermöglicht es, ein Verzeichnis auf einem entfernten System über SSH lokal als Dateisystem einzubinden.

Technisch verwendet SSHFS dafür das:

SSH File Transfer Protocol – SFTP

und stellt die entfernten Dateien über:

FUSE – Filesystem in Userspace

im lokalen Verzeichnisbaum bereit.

Dadurch können Programme auf JUMBO BLOCK Rescue auf ein entferntes Verzeichnis ähnlich zugreifen wie auf ein lokal eingehängtes Dateisystem.

Einsatzgebiet im Rescue-System

SSHFS eignet sich innerhalb von JUMBO BLOCK Rescue insbesondere für:

kontrollierten Zugriff auf Dateien eines entfernten Linux-/Unix-Systems,

Einbindung eines Remote-Verzeichnisses als lokalen Mountpunkt,

Kopieren von Diagnose- und Logdateien,

Sichern ausgewählter Remote-Dateien auf lokale Datenträger,

Bereitstellung eines entfernten Sicherungsziels,

read-only-orientierte Untersuchung entfernter Verzeichnisbestände

und Zugriff auf Dateien mit lokalen Werkzeugen, ohne sie vorher einzeln mit `scp` oder `sftp` übertragen zu müssen.

SSHFS baut auf SSH auf

SSHFS ersetzt SSH nicht.

Es verwendet die bereits in A.27 beschriebenen SSH-Mechanismen.

Dazu gehören insbesondere:

verschlüsselte Verbindung,

Host-Key-Prüfung,

Benutzer-Authentisierung,

SSH-Konfiguration

und SFTP.

Die grundlegenden Sicherheitsregeln aus A.27 gelten deshalb auch für SSHFS.

SSHFS ist kein eigener Serverdienst

SSHFS ist auf JUMBO BLOCK Rescue die Clientseite.

Das entfernte System muss einen geeigneten SSH-/SFTP-Zugang bereitstellen.

SSHFS selbst macht aus JUMBO BLOCK Rescue keinen SSH-Dateiserver.

SFTP statt Remote-Shell

SSHFS benötigt für den normalen Dateisystemzugriff das SFTP-Subsystem des entfernten SSH-Systems.

Es bindet damit einen entfernten Dateibestand ein.

Das ist etwas anderes als:

```
ssh USER@HOST
```

mit einer interaktiven Remote-Shell.

Grundsyntax

Die dokumentierte SSHFS-Syntax lautet:

```
sshfs [user@]host:[dir] mountpoint [options]
```

Ein einfaches Beispielprinzip lautet:

```
sshfs USER@HOST:/REMOTE/PFAD /LOKALER/MOUNTPUNKT
```

Beispiel

```
sshfs admin@server.example:/var/log ~/remote-logs
```

Danach erscheint das entfernte Verzeichnis:

```
/var/log
```

unter dem lokalen Mountpunkt:

```
~/remote-logs
```

Lokaler Mountpunkt

Der lokale Mountpunkt muss vorhanden und für den verwendeten Benutzer geeignet sein.

Beispielsweise:

```
mkdir ~/remote-logs
```

Anschließend:

```
sshfs admin@server.example:/var/log ~/remote-logs
```

SSHFS wird laut Projektdokumentation normalerweise als regulärer Benutzer und nicht als `root` ausgeführt; der Mountpunkt muss dafür dem Benutzer gehören.

Benutzername weglassen

Wird kein Remote-Benutzer angegeben, verwendet SSHFS standardmäßig den lokalen Benutzernamen für die Verbindung.

Deshalb ist in administrativen Rescue-Situationen die explizite Form häufig eindeutiger:

```
USER@HOST
```

Remote-Verzeichnis weglassen

Wird nach dem Doppelpunkt kein Remote-Verzeichnis angegeben, wird standardmäßig das Home-Verzeichnis des Remote-Benutzers verwendet.

Beispiel:

```
sshfs admin@server.example: ~/remote
```

Gesamtes Root-Dateisystem

Ein autorisierter Benutzer kann – sofern seine Remote-Berechtigungen dies zulassen – auch einen Pfad wie:

```
/
```

als Ausgangspunkt angeben.

Beispielprinzip:

```
sshfs USER@HOST:/ /MOUNTPUNKT
```

Das bedeutet jedoch nicht, dass dadurch automatisch uneingeschränkter Zugriff auf sämtliche Dateien des Remote-Systems entsteht.

Entscheidend bleiben die Rechte des authentisierten Remote-Benutzers.

Remote-Berechtigungen bleiben maßgebend

SSHFS umgeht keine Berechtigungen des entfernten Systems.

Welche Operationen auf dem Remote-Dateisystem tatsächlich zulässig sind, hängt von den Berechtigungen ab, die der verwendete Remote-Zugang besitzt.

Wichtige Besonderheit lokaler Rechte

SSHFS ignoriert standardmäßig lokale Dateiberechtigungsprüfungen auf der FUSE-Seite.

Entscheidend für die tatsächliche Remote-Operation ist, was der entfernte Server für die verwendeten Zugangsdaten erlaubt.

`default_permissions`

Mit:

```
-o default_permissions
```

kann zusätzlich die lokale Berechtigungsprüfung aktiviert werden.

Das ersetzt jedoch nicht die Berechtigungsprüfung des Remote-Servers.

Es ergänzt sie auf der lokalen Seite.

Standardmäßig nur für den mountenden Benutzer

Standardmäßig kann nur der Benutzer, der das SSHFS-Dateisystem eingehängt hat, darauf zugreifen.

Das ist für ein Rescue-System eine sinnvolle Sicherheitsgrenze.

`allow_other`

FUSE kann über:

```
allow_other
```

den Zugriff für weitere lokale Benutzer ermöglichen.

Damit wird der lokale Zugriffskreis erweitert.

Für einen Rescue-Einsatz sollte dies nur verwendet werden, wenn es tatsächlich erforderlich ist.

Grundregel

So wenig lokale Benutzer wie möglich auf den Remote-Mount zugreifen lassen.

Host-Key-Prüfung

Da SSHFS SSH verwendet, gelten die Host-Key-Regeln aus A.27.

Bei der ersten Verbindung muss die Identität des entfernten SSH-Servers geprüft werden.

Ein Host-Key-Fingerprint sollte bei sicherheitskritischen Systemen über eine unabhängige vertrauenswürdige Quelle bestätigt werden.

Host-Key-Warnung

Eine Warnung über einen geänderten Host Key darf auch bei SSHFS nicht einfach umgangen werden.

Mögliche Ursachen sind:

Server neu installiert,

Host Keys bewusst geändert,

System ersetzt

oder eine falsche beziehungsweise manipulierte Gegenstelle.

Erst nach Klärung darf der gespeicherte Host-Key-Zustand angepasst werden.

SSH-Optionen verwenden

SSHFS kann zahlreiche SSH-Optionen an die zugrunde liegende SSH-Verbindung weitergeben.

Dadurch können beispielsweise:

Port,

Identity-Datei,

Keepalive-Einstellungen

oder andere SSH-Verbindungsparameter

konfiguriert werden.

Abweichender SSH-Port

Ein abweichender Port kann beispielsweise mit:

```
-o port=PORT
```

angegeben werden.

Beispiel:

```
sshfs -o port=2222 USER@HOST:/PFAD /MOUNTPUNKT
```

Bestimmten privaten Schlüssel verwenden

Da SSHFS SSH verwendet, kann eine bestimmte Identity-Datei über die entsprechende SSH-Option ausgewählt werden.

Beispielprinzip:

```
sshfs -o IdentityFile=/PFAD/ZUM/SCHLUESSEL USER@HOST:/PFAD
/MOUNTPUNKT
```

Die Sicherheitsregeln für private SSH-Schlüssel aus A.27 gelten unverändert.

Read-only-Mount

Für Rescue-Arbeiten ist besonders wichtig:

```
-o ro
```

Damit wird das SSHFS-Dateisystem read-only eingebunden.

Beispielprinzip:

```
sshfs -o ro USER@HOST:/PFAD /MOUNTPUNKT
```

Bedeutung für JUMBO BLOCK Rescue

Wenn ein entferntes System untersucht werden soll und keine Veränderung erforderlich ist, sollte grundsätzlich erwogen werden:

read-only mounten.

Damit wird das Risiko versehentlicher Dateiänderungen über den SSHFS-Mount reduziert.

Read-only schützt nur diesen Zugriffspfad

Ein SSHFS-Read-only-Mount macht das entfernte System nicht insgesamt unveränderlich.

Andere Benutzer,

andere SSH-Sitzungen,

lokale Prozesse auf dem Server

oder andere Netzwerkdienste

können die Remote-Dateien weiterhin verändern.

Kein forensischer Snapshot

Ein SSHFS-Mount ist daher kein eingefrorenes Abbild eines entfernten Dateisystems.

Während der Untersuchung können sich Dateien ändern.

Das ist besonders wichtig bei:

Logs,

Datenbanken,

Mailspools,

temporären Dateien

und laufenden Anwendungen.

Live-System bleibt live

Wird ein laufender Server über SSHFS untersucht, können Prozesse des Servers gleichzeitig:

Dateien schreiben,

Dateien löschen,

Dateien rotieren,

Dateien umbenennen

oder Metadaten verändern.

Die lokale Ansicht über SSHFS kann deshalb zeitlich veränderlich sein.

Konsequenz für Beweissicherung

Für eine streng forensische Sicherung ist ein SSHFS-Mount allein nicht ausreichend.

SSHFS ist primär ein Remote-Dateizugriffswerkzeug.

Eine beweissichere Acquisition erfordert einen entsprechend kontrollierten und dokumentierten Sicherungsprozess.

Dateien kopieren

Nach dem Mount können normale lokale Dateiprogramme auf den Mountpunkt zugreifen.

Beispielsweise können Dateien vom SSHFS-Mount auf einen lokalen Rescue-Datenträger kopiert werden.

Wichtige Perspektive

Ein lokaler Befehl wie:

```
cp ~/remote-logs/system.log /media/rescue/
```

liest in diesem Beispiel über SSHFS tatsächlich Daten vom entfernten Server.

Die Quelle sieht lokal aus, liegt physisch aber remote.

Schreiben wirkt remote

Noch wichtiger ist die Gegenrichtung.

Ein lokaler Befehl wie:

```
rm ~/remote/datei
```

kann bei einem read-write eingebundenen SSHFS-Dateisystem eine Datei auf dem entfernten System löschen.

Rescue-Grundregel

Ein SSHFS-Mount darf niemals mit einem ungefährlichen lokalen Arbeitsverzeichnis verwechselt werden.

Operationen innerhalb des Mountpunkts können reale Änderungen auf dem Remote-System auslösen.

Grafische Dateimanager

Da SSHFS das Remote-Verzeichnis als lokales Dateisystem bereitstellt, können grundsätzlich auch lokale Dateimanager auf den Mountpunkt zugreifen.

Dadurch sind beispielsweise:

Kopieren,

Öffnen,

Umbenennen

oder Löschen

über eine grafische Oberfläche möglich, soweit Mountmodus und Remote-Berechtigungen dies erlauben.

Gefahr durch grafische Programme

Auch grafische Programme können unbeabsichtigte Schreibzugriffe erzeugen.

Beispielsweise durch:

Speichern,

Umbenennen,

Löschen

oder Erzeugen zusätzlicher Dateien.

Für eine reine Untersuchung ist deshalb ein read-only SSHFS-Mount besonders sinnvoll.

Programme auf Remote-Dateien anwenden

Lokale Programme können Dateien aus einem SSHFS-Mount lesen.

Damit lassen sich beispielsweise:

Textdateien betrachten,

Logs analysieren,

Hashes berechnen

oder Dateien zur weiteren Untersuchung auf ein lokales Medium kopieren.

Netzwerk bleibt Teil des Dateizugriffs

Jeder Zugriff auf eine noch nicht lokal verfügbare Datei erfordert Kommunikation mit dem Remote-System.

SSHFS macht ein Netzwerkdateisystem optisch lokal.

Es macht es technisch nicht zu einem lokalen Datenträger.

Performance

Die Leistung hängt unter anderem ab von:

Netzwerkbandbreite,

Latenz,

SSH-Verschlüsselung,

Remote-Server,

Dateigrößen,

Anzahl der Dateioperationen

und SSHFS-/FUSE-Verhalten.

Ein Verzeichnis mit sehr vielen kleinen Dateien kann sich deshalb anders verhalten als die Übertragung einer einzelnen großen Datei.

Verbindung unterbrochen

Eine wichtige Besonderheit von SSHFS betrifft Verbindungsabbrüche.

Die Dokumentation weist darauf hin, dass Dateisystemoperationen bei unterbrochener Verbindung blockieren können.

Programme, die auf den Mountpunkt zugreifen, können dadurch scheinbar „einfrieren“.

Ursache

SSHFS folgt dabei grundsätzlich dem SSH-Verhalten ohne automatisch erzwungene Netzwerk-Timeouts.

Wird eine Verbindung unbemerkt unterbrochen, können laufende Zugriffe daher warten.

`ServerAliveInterval`

Die SSHFS-Dokumentation empfiehlt für entsprechende Situationen beispielsweise:

```
-o ServerAliveInterval=15
```

Dadurch sendet die zugrunde liegende SSH-Verbindung regelmäßig Keepalive-Nachrichten.

Beispiel

```
sshfs -o ServerAliveInterval=15 USER@HOST:/PFAD /MOUNTPUNKT
```

`reconnect`

Mit:

```
-o reconnect
```

kann SSHFS versuchen, eine unterbrochene Verbindung automatisch wiederherzustellen.

Beispiel:

```
sshfs -o reconnect USER@HOST:/PFAD /MOUNTPUNKT
```

Kombination

Ein typisches robustes Verbindungsmuster kann lauten:

```
sshfs -o reconnect,ServerAliveInterval=15 USER@HOST:/PFAD
/MOUNTPUNKT
```

Wichtige Einschränkung von `reconnect`

Dateien, die bereits vor der Wiederverbindung geöffnet waren, können nach dem Reconnect Fehler liefern und müssen erneut geöffnet werden.

Kein Schutz vor Datenverlust während eines Abbruchs

Eine automatische Wiederverbindung bedeutet nicht, dass eine gerade laufende Dateioperation nahtlos und verlustfrei fortgesetzt wird.

Die SSHFS-Dokumentation warnt ausdrücklich, dass Daten, die während einer Unterbrechung gerade gelesen oder geschrieben wurden, verloren gehen können.

Bedeutung für Rescue-Kopien

Bei wichtigen Dateien sollte nach einer instabilen Verbindung nicht allein darauf vertraut werden, dass die Kopie vollständig ist.

Eine anschließende Integritätskontrolle ist sinnvoll.

Hashprüfung

Nach einer kritischen Dateiübertragung kann beispielsweise ein kryptografischer Hash der lokalen Kopie mit einem unabhängig auf der Quelle ermittelten Hash verglichen werden.

Dabei muss berücksichtigt werden, ob die Quelldatei während der Sicherung verändert werden konnte.

`delay_connect`

Mit:

```
-o delay_connect
```

wird die SSH-Verbindung nicht sofort beim Mounten aufgebaut.

Die Verbindung erfolgt erst, wenn der Mountpunkt erstmals tatsächlich angesprochen wird.

`sshfs_sync`

Mit:

```
-o sshfs_sync
```

können synchrone Schreibvorgänge aktiviert werden.

Dies kann die Verarbeitung verlangsamen.

Für einen reinen read-only Rescue-Mount spielt diese Schreiboption naturgemäß keine zentrale Rolle.

no_readahead

Mit:

```
-o no_readahead
```

liest SSHFS nur die tatsächlich angeforderten Daten, anstatt zusätzliche Daten vorausschauend einzulesen.

Dies kann in speziellen Diagnose- oder Netzwerkbedingungen relevant sein.

sync_readdir

Mit:

```
-o sync_readdir
```

wird das Lesen von Verzeichnisinhalten synchron ausgeführt.

Auch diese Option kann die Verarbeitung verlangsamen.

Caching

SSHFS verwendet Caching-Mechanismen.

Damit muss bei einem sich gleichzeitig verändernden Remote-Dateisystem berücksichtigt werden, dass die lokale Sicht nicht in jedem Moment einer transaktionsartigen Echtzeitaufnahme entspricht.

SSHFS ist kein Cluster-Dateisystem

SSHFS ist nicht dafür ausgelegt, komplexe Konsistenzgarantien für mehrere gleichzeitig schreibende Systeme bereitzustellen.

Für Rescue-Arbeiten sollte es als praktischer Remote-Dateizugriff verstanden werden.

Remote-Datei wird während des Lesens geändert

Wird eine Datei auf dem Server gleichzeitig verändert, während sie über SSHFS gelesen oder kopiert wird, kann die resultierende Kopie unter Umständen keinen konsistenten Zeitpunkt repräsentieren.

Besonders problematisch sind:

aktive Datenbanken,

virtuelle Maschinen,

Maildatenbanken,

große Logs

und andere kontinuierlich beschriebene Dateien.

Datenbankdateien

Eine laufende Datenbank sollte nicht allein dadurch gesichert werden, dass ihre aktiven Dateien über SSHFS kopiert werden.

Ob eine solche Kopie konsistent ist, hängt von der jeweiligen Datenbank und ihrem Sicherungsverfahren ab.

Für produktive Datenbanken sind die vorgesehenen Backup-/Snapshot-Verfahren zu verwenden.

Unmount

Nach Abschluss der Arbeit muss das SSHFS-Dateisystem sauber ausgehängt werden.

Für FUSE3 lautet die dokumentierte Linux-Syntax:

```
fusermount3 -u MOUNTPUNKT
```

Baseline und `fusermount3`

Die JUMBO BLOCK Rescue Baseline enthält:

`fuse3 3.17.2-3`.

Damit ist die FUSE3-Basis des installierten SSHFS-Stacks belegt.

Beispiel

```
fusermount3 -u ~/remote-logs
```

Danach ist das entfernte Verzeichnis nicht mehr über diesen Mountpunkt eingebunden.

Vor dem Unmount

Programme sollten keine Dateien innerhalb des SSHFS-Mounts mehr geöffnet haben.

Auch ein Terminal sollte sich für einen sauberen Ablauf nicht mehr mit seinem aktuellen Arbeitsverzeichnis innerhalb des Mountpunkts befinden.

„Device or resource busy“

Kann ein Mount nicht ausgehängt werden, wird er möglicherweise noch von einem Prozess verwendet.

Dann sollte zunächst festgestellt werden, welcher Prozess den Mount noch benutzt.

Nicht sofort mit Gewalt unmounten.

Verbindungsabbruch und blockierte Prozesse

Bei einem abrupten Netzwerkverlust können Prozesse, die auf SSHFS-Dateien zugreifen, blockieren.

Die SSHFS-Dokumentation weist darauf hin, dass selbst ein erzwungenes Lazy-Unmount mit `fusermount -zu` in einer solchen Situation blockierte Lese-/Schreiboperationen nicht zwingend beendet.

Konsequenz

Für länger laufende Rescue-Verbindungen sind:

`ServerAliveInterval`

und gegebenenfalls:

`reconnect`

wichtiger als die Annahme, ein späteres Force-Unmount werde jedes Problem zuverlässig beseitigen.

Read-only Rescue-Workflow

Ein konservativer Workflow lautet:

1. Remote-Ziel eindeutig identifizieren

2. SSH-Erreichbarkeit prüfen

3. Host-Key-Fingerprint verifizieren

4. geeigneten Remote-Benutzer auswählen

5. lokalen Mountpunkt erstellen

6. Remote-Verzeichnis read-only einbinden

```
sshfs -o ro USER@HOST:/PFAD /MOUNTPUNKT
```

7. benötigte Dateien untersuchen beziehungsweise auf separates Rescue-Medium kopieren

8. Integrität kritischer Kopien kontrollieren

9. alle Programme aus dem Mountpunkt herausführen

10. Mount sauber aushängen

```
fusermount3 -u /MOUNTPUNKT
```

Workflow bei instabiler Netzwerkverbindung

Falls Verbindungsabbrüche wahrscheinlich sind:

```
sshfs -o ro,reconnect,ServerAliveInterval=15 USER@HOST:/PFAD
/MOUNTPUNKT
```

Dabei bleibt zu beachten:

Ein Reconnect garantiert keine verlustfreie Fortsetzung einer gerade laufenden Dateioperation.

Workflow: Logs eines Remote-Servers sichern

1. Hostidentität prüfen

2. Remote-Logverzeichnis read-only mounten

3. benötigte Logs auf lokales Rescue-Medium kopieren

4. lokale Kopien dokumentieren beziehungsweise hashen

5. SSHFS-Mount sauber trennen

Dadurch können die Originaldateien auf dem Server weitgehend unverändert bleiben, soweit keine anderen Prozesse sie verändern.

Workflow: Remote-Sicherungsziel

SSHFS kann auch verwendet werden, um ein entferntes Verzeichnis als Ziel für gerettete Dateien einzubinden.

Dann ist der Mount bewusst read-write.

Vorher müssen jedoch geprüft werden:

verfügbarer Speicherplatz,

Remote-Berechtigungen,

Netzwerkstabilität,

Vertrauenswürdigkeit des Servers

und Datenschutzanforderungen.

Sensible Rescue-Daten

Werden sensible gerettete Daten über SSHFS auf ein entferntes System übertragen, schützt SSH die Daten auf dem Transportweg.

Nach dem Schreiben liegen sie jedoch entsprechend der Konfiguration des Zielsystems dort gespeichert vor.

Transportverschlüsselung ist keine Speicherverschlüsselung

SSH schützt die Netzwerkverbindung.

SSHFS garantiert nicht, dass die Daten auf dem Remote-Datenträger verschlüsselt gespeichert werden.

Falls dies erforderlich ist, muss das Zielsystem selbst geeignete Speicherverschlüsselung bereitstellen oder die Dateien müssen vor der Übertragung beispielsweise mit GnuPG verschlüsselt werden.

Kombination mit GnuPG

Ein möglicher Workflow lautet:

Rescue-Dateien lokal erzeugen

↓

mit GnuPG für den berechtigten Empfänger verschlüsseln

↓

verschlüsselte Dateien über SSHFS auf Remote-Speicher übertragen

Damit werden zwei verschiedene Schutzebenen kombiniert:

SSH

Transportverschlüsselung.

GnuPG

Dateiverschlüsselung.

SSHFS und SCP

Beide verwenden SSH-basierte Übertragung, erfüllen aber unterschiedliche Bedienmodelle.

SCP

kopiert gezielt Dateien zwischen zwei Orten.

SSHFS

bindet ein Remote-Verzeichnis dauerhaft in den lokalen Verzeichnisbaum ein.

Wann SCP einfacher ist

Soll lediglich:

eine einzelne Datei

oder:

ein klar definierter Verzeichnisbestand

einmalig übertragen werden, kann `scp` einfacher und übersichtlicher sein.

Wann SSHFS sinnvoll ist

SSHFS ist besonders nützlich, wenn:

mehrere lokale Werkzeuge wiederholt auf Remote-Dateien zugreifen sollen,

Verzeichnisse durchsucht werden müssen

oder ein entfernter Dateibaum für die Dauer einer Analyse lokal verfügbar sein soll.

SSHFS und SFTP

SSHFS verwendet SFTP als zugrunde liegendes Dateiübertragungsprotokoll.

Der OpenSSH-Client `sftp` stellt dagegen eine interaktive Dateiübertragungsschnittstelle bereit.

SSHFS übersetzt diese Remote-Dateioperationen in ein lokales Dateisystemmodell.

SSHFS und Midnight Commander

Das später dokumentierte:

Midnight Commander

kann einen SSHFS-Mount wie einen lokalen Verzeichnisbaum verwenden.

Damit können Remote-Dateien über eine Dateimanager-Oberfläche untersucht und kopiert werden.

Bei read-write Mounts gilt jedoch auch dort:

Löschen, Umbenennen und Überschreiben können reale Remote-Operationen auslösen.

SSHFS und ClamAV

Dateien innerhalb eines SSHFS-Mounts können grundsätzlich von lokalen Werkzeugen gelesen werden.

Ein Virenscan über einen Remote-Mount erzeugt jedoch Netzwerkzugriffe und kann erhebliche Datenmengen über die SSH-Verbindung lesen.

Für große Bestände muss dies bewusst berücksichtigt werden.

SSHFS und Hashberechnung

Auch ein lokaler Hashbefehl auf einer Remote-Datei liest deren vollständigen Inhalt über das Netzwerk.

Bei einer 50-GB-Datei müssen für einen vollständigen Hash entsprechend große Datenmengen übertragen werden.

„Lokal gemountet“ bedeutet nicht „lokal gespeichert“.

SSHFS und Netzwerkdiagnose

Bei Problemen können die bereits dokumentierten Werkzeuge kombiniert werden:

Dig

Namensauflösung prüfen.

MTR

Netzwerkpfad untersuchen.

Nmap

SSH-Port am autorisierten Ziel prüfen.

SSH

direkte Anmeldung testen.

SSHFS

anschließend Dateisystemzugriff herstellen.

Diagnosegrundsatz

Wenn SSHFS nicht funktioniert, sollte zunächst geprüft werden, ob eine normale SSH-/SFTP-Verbindung zum selben Ziel funktioniert.

Damit lässt sich unterscheiden zwischen:

grundlegendem SSH-/Netzwerkproblem

und:

SSHFS-/FUSE-spezifischem Problem.

IPv6-Adresse

Bei einer numerischen IPv6-Adresse verlangt die SSHFS-Syntax, dass die Adresse in eckige Klammern gesetzt wird.

Dies verhindert Mehrdeutigkeiten mit dem Doppelpunkt, der in der SSHFS-Pfadsyntax verwendet wird.

`/etc/fstab`

SSHFS kann grundsätzlich auch über `/etc/fstab` eingebunden werden.

Die dokumentierte Dateisystemart lautet:

`sshfs`

wobei aus Kompatibilitätsgründen auch:

`fuse.sshfs`

unterstützt wird.

Rescue-Einordnung von `/etc/fstab`

Für ein temporäres Rescue-System sind manuell und bewusst aufgebaute Mounts häufig transparenter als dauerhaft konfigurierte Remote-Mounts.

Eine permanente `/etc/fstab`-Konfiguration ist deshalb kein notwendiger Bestandteil eines normalen Rescue-Workflows.

Version 3.7.3

Die installierte Baseline-Version 3.7.3 wurde am 26. Mai 2022 veröffentlicht und enthielt laut Release-Historie kleinere Fehlerkorrekturen.

Für dieses Handbuch ist ausschließlich die in Baseline 20261001 tatsächlich installierte Version maßgebend.

Versionsgrenze

Spätere SSHFS-Versionen können zusätzliche Optionen, Sicherheitsänderungen und Fehlerkorrekturen enthalten.

Diese werden nicht rückwirkend als Funktionen von SSHFS 3.7.3 dokumentiert.

Grenzen

SSHFS ist kein lokales Dateisystem.

SSHFS ist kein Backupverfahren an sich.

SSHFS erzeugt keinen konsistenten Snapshot eines laufenden Remote-Systems.

SSHFS umgeht keine Remote-Berechtigungen.

SSHFS garantiert keine unterbrechungsfreie Netzwerkverbindung.

`reconnect` garantiert keine verlustfreie Fortsetzung bereits geöffneter Dateien.

SSHFS garantiert keine Speicherverschlüsselung auf dem Remote-System.

Ein read-only SSHFS-Mount verhindert nicht, dass andere Prozesse das Remote-Dateisystem verändern.

SSHFS ersetzt keine forensische Acquisition.

SSHFS macht ein möglicherweise kompromittiertes Remote-System nicht vertrauenswürdig.

Sicherheitshinweise

Host-Key-Fingerprint vor sicherheitskritischen Erstverbindungen prüfen.

Host-Key-Warnungen nicht ignorieren.

Private SSH-Schlüssel schützen.

Keine unnötigen privaten Schlüssel auf Remote-Systeme kopieren.

Bei kompromittierungsverdächtigen Systemen kein Agent Forwarding verwenden.

Für reine Analyse nach Möglichkeit read-only mounten.

Immer bedenken, dass lokale Dateioperationen innerhalb des Mountpunkts Remote-Operationen auslösen können.

Keine Dateien versehentlich löschen, umbenennen oder überschreiben.

Sensible Rescue-Daten nur auf autorisierte Remote-Ziele übertragen.

Transportverschlüsselung nicht mit Speicherverschlüsselung verwechseln.

Bei instabilen Verbindungen kopierte Daten anschließend auf Integrität prüfen.

Mounts nach Abschluss sauber mit `fusermount3 -u` trennen.

Technische Einordnung in JUMBO BLOCK Rescue

SSHFS schließt innerhalb von JUMBO BLOCK Rescue den Bereich:

Remote.

Die beiden Programme dieses Bereichs ergänzen sich unmittelbar:

A.27 SSH

stellt die sichere Remote-Verbindung, Authentisierung, Remote-Shell und Dateiübertragung bereit.

A.28 SSHFS

nutzt SSH/SFTP, um einen entfernten Verzeichnisbaum als lokales FUSE-Dateisystem einzubinden.

Der typische technische Ablauf lautet:

Dig / MTR / Nmap

Netzwerk und Erreichbarkeit prüfen.

↓

SSH

Hostidentität und Benutzerzugang prüfen.

↓

SSHFS

autorisiertes Remote-Verzeichnis lokal einbinden.

↓

Read-only

bei reiner Untersuchung Schreibzugriffe verhindern.

↓

lokale Rescue-Werkzeuge

Dateien untersuchen oder auf separates Medium kopieren.

↓

Integritätskontrolle

kritische Sicherungen überprüfen.

↓

fusermount3

Remote-Dateisystem sauber aushängen.

Damit entsteht ein besonders nützlicher Rescue-Mechanismus:

Ein entferntes Dateisystem kann für lokale Werkzeuge verfügbar gemacht werden, ohne dass sämtliche Dateien zunächst manuell einzeln übertragen werden müssen.

Die entscheidende Sicherheitsregel lautet:

Ein SSHFS-Verzeichnis sieht lokal aus – die darin ausgeführten Dateioperationen wirken aber auf ein entferntes reales System.

Für reine Rescue-Analyse gilt deshalb:

Remote eindeutig authentisieren, möglichst read-only mounten, benötigte Daten sichern und den Mount anschließend sauber trennen.

A.29 HWInfo

Bereich: Hardware
Programm: HWInfo
Baseline 20261001: HWInfo 21.82
Installiertes Paket: `hwinfo 21.82-1.1+b2`
Zugehörige Bibliothek: `libhd21t64 21.82-1.1+b2`
Bedienung: Kommandozeile
Aufgabe: Hardware erkennen, inventarisieren und für Diagnose- und Supportzwecke dokumentieren

Zweck

HWInfo ist ein Hardware-Erkennungswerkzeug für Linux.

Das Programm untersucht die vom laufenden System erkennbare Hardware und kann daraus eine detaillierte Übersicht erzeugen.

Die Debian-Paketbeschreibung nennt unter anderem Informationen zu:

Festplatten und Partitionen,

CD-ROM- und anderen Laufwerken,

Netzwerkkarten,

Grafikkarten,

Monitoren,

Kameras,

Mäusen,

Soundhardware,

BIOS,

CPU,

USB-Geräten,

Speicher

und Multiprozessorsystemen.

HWInfo kann außerdem einen Systemübersichts-Log erzeugen, der anschließend beispielsweise für Diagnose- und Supportzwecke verwendet werden kann.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue dient HWInfo insbesondere der Frage:

Welche Hardware erkennt das gestartete Rescue-System tatsächlich?

Typische Einsatzfälle sind:

Hardwarebestand eines unbekannten Rechners erfassen,

CPU und Arbeitsspeicher identifizieren,

Datenträger erkennen,

Partitionen erfassen,

Storage-Controller untersuchen,

PCI-Geräte auflisten,

USB-Geräte identifizieren,

Netzwerkhardware feststellen,

Grafikhardware erkennen,

Monitorinformationen untersuchen,

BIOS-bezogene Informationen erfassen,

Soundhardware identifizieren

und einen Hardwarebericht für die weitere Diagnose erstellen.

Hardware-Erkennung statt Gesundheitsdiagnose

HWInfo ist primär ein:

Hardware Detection Tool.

Das muss von einer Zustands- beziehungsweise Gesundheitsdiagnose unterschieden werden.

HWInfo kann beispielsweise einen Datenträger erkennen und beschreiben.

Für dessen SMART-Zustand ist jedoch:

A.8 SMART Diagnose

zuständig.

Für NVMe-spezifische Zustandsinformationen:

A.9 NVMe Diagnose.

Grundaufruf

HWInfo wird über:

`hwinfo`

aufgerufen.

Die Manpage beschreibt einen Aufruf ohne Optionen als ungefähr gleichbedeutend mit:

```
hwinfo --all --log=-
```

Damit wird eine umfangreiche Hardwareerkennung durchgeführt und die Ausgabe ausgegeben.

Umfang der Ausgabe

Eine vollständige HWInfo-Ausgabe kann sehr umfangreich werden.

Für Rescue-Arbeiten ist es deshalb häufig sinnvoller, zunächst nur die relevante Hardwareklasse abzufragen.

Hardwareklassen

HWInfo unterstützt gezielte Hardwareabfragen nach dem Schema:

`hwinfo --HARDWARE_ITEM`

Zu den dokumentierten Hardwareklassen gehören unter anderem:

`--bios`

`--block`

`--cpu`

`--disk`

`--gfxcard`

`--memory`

`--monitor`

`--netcard`

`--network`

`--partition`

`--pci`

`--scsi`

`--sound`

`--storage-ctrl`

`--usb`

`--usb-ctrl`

`--wlan`

sowie zahlreiche weitere Klassen.

Mehrere Klassen kombinieren

Hardwareklassen können mehrfach angegeben werden.

Damit lassen sich mehrere relevante Bereiche in einem Aufruf untersuchen.

Das ist beispielsweise nützlich, wenn:

Datenträger,

Storage-Controller

und:

Partitionen

gemeinsam betrachtet werden sollen.

Kurzansicht

Mit:

`--short`

erzeugt HWInfo eine kompaktere Zusammenfassung.

Die Option wird zusätzlich zu einer Hardwareklasse verwendet.

Beispiel:

```
hwinfo --short --disk
```

Schneller Systemüberblick

Für einen kompakten Überblick eignet sich beispielsweise:

```
hwinfo --short --block
```

Die HWInfo-Manpage verwendet genau diesen Aufruf als Beispiel für eine Übersicht über Blockgeräte.

CPU

Mit:

```
hwinfo --cpu
```

werden erkannte Prozessorinformationen untersucht.

Dies kann bei einem unbekannten Zielsystem eine schnelle Antwort auf die Frage geben:

Welche CPU erkennt das Rescue-System?

Arbeitsspeicher

Mit:

```
hwinfo --memory
```

können von HWInfo ermittelte Speicherinformationen ausgegeben werden.

Wichtige Abgrenzung beim Arbeitsspeicher

Hardwareerkennung ist kein RAM-Belastungstest.

Dass HWInfo Arbeitsspeicher erkennt, beweist nicht, dass sämtliche Speicherzellen fehlerfrei funktionieren.

Die Baseline enthält für weitergehende Speicherdiagnose separate Werkzeuge wie `memtest86+` und `memtester`; diese gehören jedoch nicht zu den 35 Programmen dieses Block-A-Handbuchs.

Datenträger

Mit:

```
hwinfo --disk
```

werden erkannte Datenträger untersucht.

Die Manpage nennt diesen Aufruf ausdrücklich als Beispiel:

```
hwinfo --disk
```

Einzelnen Datenträger untersuchen

HWInfo unterstützt:

```
--only DEVNAME
```

Damit kann die Ausgabe auf passende Geräte eingeschränkt werden.

Die entsprechende Hardwareklasse muss zusätzlich angegeben werden.

Dokumentiertes Beispiel:

```
hwinfo --disk --only /dev/sdb
```

Bedeutung im Rescue-Fall

Wenn mehrere Laufwerke vorhanden sind, kann eine gezielte Abfrage helfen, ein bestimmtes Gerät genauer zuzuordnen.

Das ist besonders vor Operationen wichtig, die auf Gerätepfade wie:

```
/dev/sda
```

```
/dev/sdb
```

oder:

```
/dev/nvme...
```

zugreifen.

Gerätebezeichnungen niemals allein aus einer früheren Sitzung übernehmen

Linux-Gerätebezeichnungen können von der jeweils erkannten Hardwarekonfiguration abhängen.

Vor kritischen Rescue-Operationen muss deshalb die aktuelle Zuordnung erneut geprüft werden.

HWInfo kann dabei eine zusätzliche Informationsquelle darstellen.

Blockgeräte

Mit:

```
hwinfo --block
```

werden Blockgeräte untersucht.

In der Kurzform:

```
hwinfo --short --block
```

entsteht eine kompakte Übersicht.

Partitionen

Mit:

```
hwinfo --partition
```

kann gezielt die von HWInfo erkannte Partitionshardware beziehungsweise -struktur abgefragt werden.

Storage-Controller

Mit:

```
hwinfo --storage-ctrl
```

werden Storage-Controller untersucht.

Das ist für Rescue-Arbeiten besonders wichtig, wenn ein Datenträger:

nicht erscheint,

hinter einem Controller liegt,

über RAID-/Storage-Hardware angebunden ist

oder die Verbindung zwischen Datenträger und Betriebssystem untersucht werden muss.

SCSI

Mit:

```
hwinfo --scsi
```

kann die entsprechende Geräteklasse untersucht werden.

Dabei ist zu beachten, dass die Linux-Storage-Schicht auch Geräte über SCSI-artige Schnittstellen darstellen kann, die physisch nicht einem klassischen parallelen SCSI-System entsprechen.

RAID-Geräte

Die HWInfo-Manpage weist ausdrücklich darauf hin, dass RAID-Geräte normalerweise nicht ausgegeben werden.

Mit:

```
--listmd
```

können sie zusätzlich in die Ausgabe aufgenommen werden.

Bedeutung

Wenn Software-RAID-Strukturen relevant sind, kann deshalb:

```
--listmd
```

eine wichtige Ergänzung darstellen.

Die eigentliche RAID-Analyse oder Reparatur ist jedoch nicht Aufgabe von HWInfo.

PCI-Geräte

Mit:

```
hwinfo --pci
```

werden erkannte PCI-Geräte untersucht.

Dies kann beispielsweise hilfreich sein bei:

Storage-Controllern,

Netzwerkadaptern,

Grafikcontrollern

und anderen PCI-/PCIe-Geräten.

USB-Geräte

Mit:

```
hwinfo --usb
```

werden erkannte USB-Geräte abgefragt.

Das ist in einer Rescue-Umgebung besonders nützlich, weil häufig zusätzliche Geräte angeschlossen werden:

USB-Festplatten,

USB-SSDs,

Flash-Laufwerke,

USB-Netzwerkadapter,

Adaptergehäuse

oder andere Peripherie.

USB-Controller

Mit:

```
hwinfo --usb-ctrl
```

können USB-Controller separat untersucht werden.

Damit lässt sich zwischen:

angeschlossenem USB-Gerät

und:

USB-Controller des Systems

unterscheiden.

Netzwerkhardware

Mit:

```
hwinfo --netcard
```

können erkannte Netzwerkadapter untersucht werden.

Dies hilft beispielsweise bei der Frage:

Welche Netzwerkschnittstellen beziehungsweise Netzwerkgeräte erkennt das Rescue-System hardwareseitig?

Netzwerk und Netcard

HWInfo kennt sowohl Hardwareklassen wie:

```
netcard
```

als auch:

```
network
```

.

Die Ausgabe ist deshalb gezielt nach dem jeweiligen Diagnosezweck auszuwählen.

WLAN

Mit:

```
hwinfo --wlan
```

kann speziell WLAN-bezogene Hardware untersucht werden.

Hardware erkannt, Netzwerk funktioniert trotzdem nicht

Dass ein Netzwerkadapter von HWInfo erkannt wird, beweist nicht, dass:

eine IP-Adresse vorhanden ist,

das Netzwerk erreichbar ist,

DNS funktioniert

oder eine Internetverbindung besteht.

Hierfür sind die Netzwerkwerkzeuge aus A.17 bis A.24 zuständig.

Grafikkarte

Mit:

```
hwinfo --gfxcard
```

kann die erkannte Grafikhardware untersucht werden.

Dies kann beispielsweise bei:

Darstellungsproblemen,

falscher Auflösung,

mehreren Grafikadaptern

oder Treiberproblemen

erste Hinweise liefern.

Monitor

Mit:

```
hwinfo --monitor
```

werden Monitorinformationen untersucht.

HWInfo besitzt hierfür auch spezielle Mechanismen zur Monitorerkennung.

BIOS

Mit:

```
hwinfo --bios
```

können BIOS-bezogene Informationen abgefragt werden.

Für detaillierte SMBIOS-/DMI-Strukturen besitzt JUMBO BLOCK Rescue zusätzlich:

A.32 DMI Decode – dmidecode.

Sound

Mit:

```
hwinfo --sound
```

kann erkannte Soundhardware untersucht werden.

Kamera

HWInfo unterstützt außerdem die Hardwareklasse:

`camera`.

Bluetooth

Die dokumentierten Hardwareklassen enthalten:

`bluetooth`.

Tastatur und Maus

HWInfo unterstützt unter anderem:

`keyboard`

und:

`mouse`.

Drucker und Scanner

Auch:

`printer`

und:

`scanner`

gehören zu den dokumentierten Hardwareklassen.

Damit geht HWInfo deutlich über reine PC-Kernkomponenten hinaus.

`--all`

Mit:

```
hwinfo --all
```

wird eine umfassende Hardwareerkennung ausgelöst.

`--reallyall`

Zusätzlich existiert die Hardwareklasse:

`reallyall`.

Sie gehört zur dokumentierten Liste der verfügbaren Hardware-Items.

Für normale Rescue-Arbeiten ist jedoch meist eine gezielte Abfrage vorzuziehen, da vollständige Ausgaben sehr umfangreich sein können.

Ausgabe protokollieren

Mit:

```
--log FILE
```

schreibt HWInfo Loginformationen in eine Datei.

Dabei muss zusätzlich eine Hardwareklasse angegeben werden, damit eine entsprechende Hardwareerkennung ausgelöst wird.

Beispielprinzip:

```
hwinfo --all --log hwinfo.log
```

Nutzen eines Hardwarelogs

Ein Hardwarelog kann beispielsweise:

dem Support übergeben,

einem Rescue-Vorgang beigefügt,

vor und nach einer Hardwareänderung verglichen

oder für eine spätere Analyse gespeichert

werden.

Datenschutz des Logs

Ein vollständiges Hardwarelog kann identifizierende technische Informationen enthalten.

Dazu können je nach erkannter Hardware beispielsweise gehören:

Gerätebezeichnungen,

Herstellerinformationen,

Modellinformationen,

Hardwarekennungen,

Netzwerkschnittstelleninformationen

und weitere Systemdetails.

Ein solches Log sollte deshalb vor einer öffentlichen Weitergabe geprüft werden.

Konfiguration speichern

HWInfo unterstützt:

```
--save-config SPEC
```

Damit kann die Konfiguration eines bestimmten Geräts beziehungsweise aller Geräte unter:

```
/var/lib/hardware
```

gespeichert werden.

Die Option muss zusammen mit einer Hardwareerkennung verwendet werden.

Gespeicherte Konfiguration anzeigen

Mit:

```
--show-config UDI
```

können zuvor gespeicherte Konfigurationsdaten für eine entsprechende UDI angezeigt werden.

Gerätezuordnung vergleichen

HWInfo besitzt außerdem:

```
--map
```

Wenn sich Datenträgernamen beispielsweise nach einer Kerneländerung verändert haben, kann damit eine Zuordnung ausgegeben werden.

Voraussetzung ist, dass zuvor mit:

```
--save-config
```

entsprechende Konfigurationsdaten gespeichert wurden.

Bedeutung für ein Live-Rescue-System

Die Funktionen:

```
--save-config
```

und:

```
--map
```

setzen einen vorher gespeicherten Referenzzustand voraus.

Bei einem erstmals gestarteten Rescue-System auf einem unbekannten Rechner existiert dieser Referenzzustand typischerweise noch nicht.

Für eine spontane Erstdiagnose sind daher die normalen Hardwareabfragen wichtiger.

Debug-Ausgabe

Mit:

```
--debug N
```

kann ein Debug-Level gesetzt werden.

Die Debuginformationen werden laut Manpage im Log ausgegeben. Wird eine Logdatei angegeben, wird automatisch ein sinnvoller Debug-Level gesetzt.

`--verbose`

Die Option:

`--verbose`

erhöht laut Manpage die Ausführlichkeit ausschließlich zusammen mit:

`--map`.

Sie ist daher nicht als allgemeiner „zeige alle Hardwaredetails“-Schalter zu verstehen.

Hardwaredatenbank

HWInfo verwendet eine Hardwaredatenbank.

Die Manpage nennt unter anderem:

`/var/lib/hardware/hd.ids`

als externe Hardwaredatenbank.

Datenbank ausgeben

Mit:

`--dump-db N`

kann die Hardwaredatenbank ausgegeben werden.

Dabei bezeichnet:

`0`

die externe Datenbank

und:

`1`

die interne Datenbank.

Version anzeigen

Mit:

`hwinfo --version`

wird die Version der zugrunde liegenden `libhd` ausgegeben.

In der JUMBO BLOCK Rescue Baseline sind `hwinfo` und `libhd21t64` beide in Version:

21.82-1.1+b2

vorhanden.

Root-Rechte

Einige Hardwareinformationen können abhängig von Gerät, Kernel und Zugriffsrechten privilegierten Zugriff erfordern.

Wenn eine normale Benutzerabfrage nicht alle erwarteten Informationen liefert, kann eine gezielte privilegierte Diagnose erforderlich sein.

Root nicht pauschal einsetzen

Für eine reine Hardwareübersicht sollte nicht automatisch jeder Befehl mit Root-Rechten ausgeführt werden.

Das Prinzip lautet:

nur so viele Rechte wie für die jeweilige Diagnose erforderlich.

Erkennung hängt vom laufenden Kernel ab

HWInfo arbeitet aus dem laufenden Rescue-System heraus.

Damit hängt die sichtbare Hardware auch davon ab, was:

Kernel,

Treiber,

Firmware

und Hardwareplattform

dem System zur Verfügung stellen.

„Nicht von HWInfo erkannt“ bedeutet nicht zwingend „nicht vorhanden“

Die HWInfo-Manpage nennt ausdrücklich als Einschränkung:

Nicht jede Hardware kann erkannt werden.

Ein fehlender Eintrag ist deshalb kein absoluter Beweis dafür, dass die betreffende Hardware physisch nicht existiert.

Mögliche Ursachen fehlender Erkennung

Ein Gerät kann beispielsweise:

vom Kernel nicht erkannt,

durch fehlende Unterstützung nicht korrekt angesprochen,

hinter einer anderen Hardwareebene verborgen

oder aus anderen technischen Gründen nicht vollständig sichtbar

sein.

Die konkrete Ursache muss mit weiteren Diagnosewerkzeugen untersucht werden.

HWInfo verändert keine Partitionen

Die normalen Hardware-Probe-Funktionen von HWInfo dienen der Erkennung und Ausgabe.

HWInfo ist kein Partitionierungswerkzeug.

Für Partitionsänderungen ist beispielsweise:

A.6 GParted

zuständig.

HWInfo repariert keine Datenträger

HWInfo kann Datenträger erkennen.

Es repariert jedoch weder:

Partitionstabellen,

Dateisysteme,

SMART-Fehler

noch physische Datenträgerschäden.

Typischer Rescue-Fall: unbekannter Rechner

Ein sinnvoller erster Überblick kann lauten:

```
hwinfo --short --block
```

```
hwinfo --short --cpu
```

```
hwinfo --short --netcard
```

```
hwinfo --short --gfxcard
```

Damit wird zunächst festgestellt, welche wesentlichen Komponenten das Rescue-System erkennt.

Typischer Rescue-Fall: Festplatte fehlt

Ein Datenträger wird erwartet, erscheint aber nicht wie vorgesehen.

Dann kann der Diagnoseweg beispielsweise lauten:

1. Blockgeräte prüfen

```
hwinfo --short --block
```

2. Datenträger prüfen

```
hwinfo --disk
```

3. Storage-Controller prüfen

```
hwinfo --storage-ctrl
```

4. PCI-Hardware prüfen

```
hwinfo --pci
```

5. bei USB-Anbindung USB-Hardware prüfen

```
hwinfo --usb
```

6. anschließend spezialisierte Storage-Werkzeuge einsetzen

Typischer Rescue-Fall: USB-Laufwerk wird nicht erkannt

1. USB-Geräte erfassen

```
hwinfo --usb
```

2. USB-Controller prüfen

```
hwinfo --usb-ctrl
```

3. Blockgeräte prüfen

```
hwinfo --short --block
```

4. danach Datenträger- beziehungsweise Dateisystemebene untersuchen

Damit lässt sich die Diagnose schrittweise von der Hardwareverbindung zur Speicherstruktur führen.

Typischer Rescue-Fall: Netzwerk funktioniert nicht

1. Netzwerkhardware identifizieren

```
hwinfo --netcard
```

2. gegebenenfalls WLAN-Hardware prüfen

```
hwinfo --wlan
```

3. anschließend Schnittstellen- und IP-Konfiguration untersuchen

4. danach mit Ping, Dig, MTR oder anderen Netzwerkwerkzeugen weiterarbeiten

HWInfo beantwortet dabei primär die Hardwarefrage.

Typischer Rescue-Fall: falsche Bildschirmauflösung

1. Grafikhardware prüfen

```
hwinfo --gfxcard
```

2. Monitor prüfen

```
hwinfo --monitor
```

3. anschließend Treiber-, Kernel- und Displaykonfiguration untersuchen

Typischer Rescue-Fall: Supportbericht

Ein umfangreicher Hardwarebericht kann beispielsweise mit einer gezielten Kombination von Hardwareklassen oder einer vollständigen Erfassung erzeugt werden.

Vor der Weitergabe müssen enthaltene sensible oder identifizierende Informationen geprüft werden.

HWInfo und Inxi

Das folgende Programm:

A.30 Inxi

erzeugt ebenfalls umfangreiche Systeminformationen.

Die Werkzeuge überschneiden sich teilweise, haben aber unterschiedliche Darstellungs- und Abfragemodelle.

HWInfo ist besonders auf die detaillierte Hardwareerkennung nach Geräteklassen ausgerichtet.

HWInfo und LSHW

A.31 LSHW

stellt Hardwareinformationen in einer hierarchischen Hardwarestruktur dar.

HWInfo arbeitet dagegen stark mit gezielt auswählbaren Hardwareklassen.

Beide Werkzeuge können sich bei der Diagnose gegenseitig ergänzen.

HWInfo und dmidecode

A.32 dmidecode

liest SMBIOS-/DMI-Strukturen und stellt damit firmwareseitig bereitgestellte Hardwareinformationen dar.

HWInfo verfolgt eine breitere Hardwareerkennung.

HWInfo und lm-sensors

A.33 lm-sensors

konzentriert sich auf Sensorwerte wie Temperaturen, Spannungen und Lüfterdaten, soweit die Hardware entsprechende Sensoren bereitstellt und unterstützt wird.

HWInfo ist kein Ersatz für diese laufende Sensorüberwachung.

HWInfo und SMART

HWInfo

beantwortet:

Welcher Datenträger wird erkannt?

SMART Diagnose

beantwortet:

Welche Zustands- und Fehlerdaten meldet ein unterstützter Datenträger?

Diese Fragen sind getrennt.

HWInfo und NVMe Diagnose

Analog gilt:

HWInfo kann NVMe-bezogene Hardware im Systemkontext sichtbar machen.

`nvme-cli` ist dagegen das spezialisierte Werkzeug für NVMe-Controller, Namespaces, SMART-/Health-Informationen, Fehlerlogs und weitere NVMe-Funktionen.

Hardwareerkennung vor Reparatur

Vor jeder kritischen Rescue-Operation sollte eindeutig feststehen:

welches Gerät das Ziel ist,

welches Gerät das Quellmedium ist

und:

welches Gerät das Sicherungsmedium ist.

HWInfo kann hierzu eine zusätzliche unabhängige Hardwareansicht liefern.

Keine Gerätebezeichnung raten

Bei mehreren ähnlich großen Laufwerken darf niemals allein anhand einer Vermutung entschieden werden:

„`/dev/sda` wird schon die defekte Platte sein.“

Vor einem schreibenden Vorgang müssen Geräteidentität und Rolle eindeutig geprüft werden.

Besonders kritisch bei ddrescue

Bei GNU ddrescue kann eine Verwechslung von Quelle und Ziel das falsche Medium überschreiben.

HWInfo kann deshalb zusammen mit anderen Datenträgerwerkzeugen vor dem eigentlichen Rettungsvorgang zur eindeutigen Identifikation eingesetzt werden.

Hardwarebericht vor Ausbau

Vor dem physischen Austausch einer Komponente kann ein HWInfo-Bericht den erkannten Ausgangszustand dokumentieren.

Nach der Änderung kann eine erneute Erfassung beim Vergleich helfen.

Live-System als Vorteil

Da JUMBO BLOCK Rescue unabhängig vom installierten Zielbetriebssystem gestartet wird, kann HWInfo Hardwareinformationen auch dann erfassen, wenn das installierte Betriebssystem:

nicht mehr startet,

beschädigt ist

oder seine eigene Treiber-/Systemkonfiguration fehlerhaft ist.

Live-System als Grenze

Gleichzeitig sieht HWInfo die Hardware aus Sicht des:

JUMBO BLOCK Rescue Kernels und seiner Treiber.

Das Ergebnis muss deshalb nicht vollständig mit der Sicht eines anderen Betriebssystems übereinstimmen.

Grenzen

HWInfo erkennt Hardware, repariert sie aber nicht.

HWInfo ist kein SMART-Diagnoseprogramm.

HWInfo ist kein NVMe-Spezialwerkzeug.

HWInfo ist kein RAM-Stresstest.

HWInfo ist kein CPU-Stresstest.

HWInfo ist kein Sensorüberwachungsprogramm.

HWInfo ist kein Partitionierungswerkzeug.

HWInfo ist kein Dateisystemreparaturwerkzeug.

HWInfo garantiert keine vollständige Erkennung jeder vorhandenen Hardware.

Ein erkannter Datenträger ist nicht automatisch gesund.

Ein erkanntes Netzwerkgerät bedeutet nicht automatisch funktionierende Netzwerkkommunikation.

Ein erkannter Speicherbereich bedeutet nicht automatisch fehlerfreien RAM.

Sicherheitshinweise

Vor kritischen Datenträgeroperationen Geräteidentitäten eindeutig prüfen.

Hardwareerkennung nicht mit Hardwaregesundheit verwechseln.

Bei physisch auffälligen Datenträgern keine unnötigen Vollscans oder Belastungen durchführen.

Für Datenträgerzustand SMART beziehungsweise NVMe-Diagnose verwenden.

Bei instabilen Datenträgern anschließend möglichst mit GNU ddrescue auf einer Kopie weiterarbeiten.

Hardwarelogs vor externer Weitergabe auf sensible Systeminformationen prüfen.

Root-Rechte nur verwenden, wenn sie für die konkrete Diagnose erforderlich sind.

Aus einem fehlenden HWInfo-Eintrag nicht vorschnell schließen, dass die Hardware physisch nicht vorhanden ist.

Technische Einordnung in JUMBO BLOCK Rescue

HWInfo eröffnet im JUMBO BLOCK Rescue Handbuch den Bereich:

Hardware.

Dieser Bereich besteht aus:

A.29 HWInfo

breite Hardwareerkennung nach Geräteklassen.

A.30 Inxi

kompakte bis detaillierte Systemübersicht.

A.31 LSHW

hierarchische Hardwareinventarisierung.

A.32 DMI Decode

SMBIOS-/DMI-Informationen.

A.33 Sensoren

Temperatur-, Spannungs- und Lüftersensoren.

Ein typischer Hardware-Diagnoseweg kann lauten:

HWInfo

Welche Hardware erkennt das Rescue-System?

↓

Inxi

Wie sieht die kompakte Gesamtkonfiguration aus?

↓

LSHW

Wie ist die Hardware hierarchisch aufgebaut?

↓

dmidecode

Welche System-, Board-, BIOS- und Speicherinformationen liefert SMBIOS/DMI?

↓

lm-sensors

Welche unterstützten Sensorwerte werden aktuell gemeldet?

Für Storage-Probleme folgt anschließend gegebenenfalls:

SMART / NVMe Diagnose

↓

GNU ddrescue

↓

Dateisystem- und Datenrettungswerkzeuge.

Die zentrale Rolle von HWInfo lautet damit:

Hardware sichtbar und identifizierbar machen, bevor auf ihr gearbeitet wird.

Gerade im Rescue-Betrieb ist diese Bestandsaufnahme entscheidend, weil eine Reparatur oder Datenrettung erst dann sicher geplant werden kann, wenn eindeutig bekannt ist:

Welche Hardware sieht das System tatsächlich?

A.30 Inxi

Bereich: Hardware
Programm: Inxi
Baseline 20261001: Inxi 3.3.38
Installiertes Paket: `inxi 3.3.38-1-1`
Bedienung: Kommandozeile
Aufgabe: Konsolidierte System-, Hardware- und Konfigurationsinformationen für Diagnose und Support ausgeben

Zweck

Inxi ist ein Kommandozeilenwerkzeug zur Ausgabe umfangreicher Systeminformationen.

Es fasst Informationen aus unterschiedlichen Bereichen des laufenden Systems in einer einheitlichen und gut lesbaren Darstellung zusammen.

Dazu gehören unter anderem:

System und Kernel,

Rechner beziehungsweise Mainboard,

CPU,

Arbeitsspeicher,

Grafik,

Audio,

Netzwerk,

Datenträger,

Partitionen,

RAID,

USB,

Batterie,

Sensoren,

Treiber

und weitere Systeminformationen.

Inxi wird ausdrücklich auch als Diagnosewerkzeug für technischen Support eingesetzt, um die Systemkonfiguration eines Rechners schnell zu erfassen.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich Inxi insbesondere für:

schnellen Gesamtüberblick über ein unbekanntes System,

Dokumentation der vom Rescue-System erkannten Konfiguration,

Identifikation von CPU, Grafik, Netzwerk und Storage,

Ermittlung verwendeter Kernel-Treiber,

Übersicht über Datenträger und Partitionen,

Erfassung von RAM-Informationen,

Anzeige unterstützter Sensorinformationen,

Diagnose von Grafik- und Netzwerkproblemen

und Erstellung eines für Supportzwecke geeigneten Systemberichts.

Unterschied zu HWInfo

A.29 HWInfo und Inxi überschneiden sich teilweise, verfolgen jedoch unterschiedliche Schwerpunkte.

HWInfo

ist stark auf die gezielte Hardwareerkennung nach Geräteklassen ausgerichtet.

Inxi

erstellt einen zusammenhängenden Systembericht, in dem Hardware, Treiber, Kernel, Desktop-Umgebung und weitere Systeminformationen gemeinsam dargestellt werden.

Damit beantwortet HWInfo eher:

Welche Hardware erkennt das System?

Inxi beantwortet zusätzlich:

Wie sieht die aktuelle Systemkonfiguration als Ganzes aus?

Grundaufruf

Der einfachste Aufruf lautet:

```
inxi
```

Ohne weitere Optionen erzeugt Inxi einen kurzen Systembericht.

Basisbericht

Mit:

```
inxi -b
```

wird ein grundlegender Bericht erzeugt.

Dieser enthält unter anderem Informationen zu:

System,

Maschine,

Batterie, falls vorhanden,

CPU,

Grafik,

Netzwerkgeräten,

Datenträgern

und allgemeinen Systeminformationen.

Bedeutung für Rescue

`inxi -b`

ist ein geeigneter Einstieg, wenn zunächst eine kompakte Übersicht benötigt wird, ohne sofort einen sehr umfangreichen Bericht zu erzeugen.

Erweiterter Bericht

Die 3.3.x-Generation unterstützt einen umfangreicheren Gesamtbericht.

Traditionell wurde hierfür häufig:

`inxi -F`

verwendet.

Die neuere Inxi-Dokumentation kennzeichnet `-F` inzwischen als veraltet und verwendet:

`-e`

beziehungsweise:

`--expanded`

für den erweiterten Bericht.

Für die eingefrorene Baseline ist entscheidend, welche Syntax die installierte Version 3.3.38 tatsächlich bereitstellt.

Detailstufen mit `-v`

Inxi unterstützt abgestufte Verbosity-Level:

`-v 0`

bis:

`-v 8`.

Mit zunehmender Stufe werden zusätzliche Informationsbereiche und Detailstufen aktiviert.

`-v 0`

Entspricht einem einfachen Bericht.

`-v 1`

Erweitert den Bericht unter anderem um:

System,

grundlegende CPU-Daten,

Grafik,

grundlegende Datenträgerdaten

und allgemeine Informationen.

`-v 2`

Ergänzt unter anderem:

Maschineninformationen,

Batterie, soweit vorhanden,

und Netzwerkgeräte.

Diese Stufe entspricht dem grundlegenden Bericht:

`inxi -b`.

Höhere Detailstufen

Mit zunehmender Verbosity können unter anderem hinzukommen:

vollständigere CPU-Daten,

Netzwerkinformationen,

Datenträger,

Partitionen,

Arbeitsspeicher,

Audio,

Bluetooth,

RAID,

Swap,

Sensoren,

USB,

CPU-Flags

und weitere administrative Informationen.

Rescue-Regel

Nicht automatisch den maximal möglichen Bericht erzeugen.

Besser ist:

zuerst kompakt → danach gezielt detaillierter.

Das erleichtert die Auswertung und reduziert unnötig umfangreiche Diagnoseausgaben.

CPU

Mit:

```
inxi -C
```

werden CPU-Informationen ausgegeben.

Je nach verfügbaren Daten können dazu unter anderem gehören:

CPU-Modell,

Topologie,

Kerne,

Threads,

Cacheinformationen

und Taktinformationen.

CPU-Flags

Mit:

```
inxi -f
```

können CPU-Flags beziehungsweise CPU-Features ausführlicher ausgegeben werden.

Diese Ausgabe kann sehr umfangreich werden.

Für eine normale Hardwareinventarisierung ist sie nicht immer erforderlich.

CPU-Erkennung ist kein CPU-Test

Inxi kann CPU-Eigenschaften darstellen.

Es führt damit aber keinen Belastungstest der CPU durch.

Eine korrekt erkannte CPU ist nicht automatisch nachweislich fehlerfrei.

Maschine

Mit:

```
inxi -M
```

werden Maschineninformationen ausgegeben.

Dazu können abhängig von der Plattform unter anderem Angaben zum:

System,

Mainboard

und Firmware-/BIOS-Kontext

gehören.

DMI-Abgrenzung

Für eine gezielte Untersuchung der SMBIOS-/DMI-Tabellen steht in JUMBO BLOCK Rescue zusätzlich:

A.32 dmidecode

zur Verfügung.

Inxi fasst entsprechende Informationen dagegen in einen größeren Systembericht ein.

Arbeitsspeicher

Mit:

```
inxi -m
```

werden detailliertere Speicherinformationen angefordert.

Welche Informationen tatsächlich verfügbar sind, hängt vom System und den zugänglichen Datenquellen ab.

RAM-Erkennung ist kein RAM-Test

Inxi zeigt Informationen über Arbeitsspeicher.

Es überprüft jedoch nicht sämtliche Speicherzellen auf Fehler.

Ein erkannter RAM-Riegel kann trotzdem defekt sein.

Grafik

Mit:

```
inxi -G
```

werden Grafikinformationen ausgegeben.

Inxi kann dabei unter anderem Informationen über:

Grafikgeräte,

Treiber,

Display-Server beziehungsweise Wayland-Komponenten,

Auflösungen

und verfügbare Grafik-APIs

zusammenführen.

Rescue-Einsatz Grafik

Dies ist besonders hilfreich bei:

schwarzem Bildschirm,

falscher Auflösung,

Treiberproblemen,

mehreren GPUs,

Wayland-/X11-Fragen

oder unbekannter Grafikhardware.

Audio

Mit:

```
inxi -A
```

werden Audioinformationen ausgegeben.

Dazu gehören erkannte Audiogeräte und zugehörige Treiber; abhängig vom System können auch aktive Audio-APIs und Soundserver erscheinen.

Netzwerkgeräte

Mit:

```
inxi -N
```

werden Netzwerkgeräte angezeigt.

Dies ist zunächst vor allem eine Hardware-/Treiberansicht.

Erweiterte Netzwerkinformationen

Mit:

```
inxi -n
```

werden weitergehende Netzwerkinformationen ausgegeben.

IP-Adressen

Inxi kann auch IP-bezogene Informationen anzeigen.

Diese Informationen können bei der Fehlerdiagnose nützlich, bei der Veröffentlichung eines Reports aber datenschutz- beziehungsweise sicherheitsrelevant sein.

Datenschutz

Inxi besitzt deshalb spezielle Filterfunktionen für Supportberichte.

Das ist eine der wichtigsten Eigenschaften des Programms für JUMBO BLOCK Rescue.

-z – sensible Angaben filtern

Mit:

```
inxi -z
```

beziehungsweise:

```
--filter
```

werden sensible Informationen gefiltert.

Die Inxi-Dokumentation nennt insbesondere:

IP-Adressen,

MAC-Adressen,

Seriennummern,

Ortsinformationen,

Benutzernamen im Home-Pfad

und Hostnamen.

Beispiel

Für einen Supportbericht kann beispielsweise ein umfangreicher Bericht mit zusätzlicher Filterung erzeugt werden.

Grundprinzip:

```
inxi -Fz
```

beziehungsweise bei entsprechender Syntax:

```
inxi -ez
```

Zentrale Support-Regel

Vor dem Weitergeben eines Inxi-Berichts sensible Informationen filtern und den fertigen Bericht trotzdem nochmals kontrollieren.

Automatische Filterung ersetzt keine abschließende Sichtprüfung.

`-Z`

Inxi besitzt außerdem eine Option zum Aufheben bestimmter automatischer Filterungen.

Diese kann bei interner Diagnose hilfreich sein, sollte aber nicht unbedacht für öffentlich weitergegebene Reports verwendet werden.

Datenträger

Mit:

```
inxi -D
```

werden Datenträgerinformationen ausgegeben.

Dazu gehören grundlegende Informationen über erkannte Laufwerke und den vorhandenen Speicher.

Datenträgerinformationen sind keine vollständige Zustandsdiagnose

Inxi kann Informationen zu einem Laufwerk darstellen.

Für eine gezielte Gesundheitsdiagnose sind jedoch:

A.8 SMART Diagnose

und:

A.9 NVMe Diagnose

die spezialisierten Werkzeuge.

Partitionen

Inxi besitzt verschiedene Optionen zur Darstellung von Partitionen und Mountzuständen.

Damit lässt sich ein Systembericht um Informationen zu:

Systempartitionen,

eingehängten Partitionen

und weiteren Speicherstrukturen

ergänzen.

RAID

Mit:

```
inxi -R
```

können RAID-bezogene Informationen angezeigt werden, soweit entsprechende Strukturen vorhanden und erkennbar sind.

RAID-Ausgabe ist keine RAID-Reparatur

Inxi kann RAID-Informationen zusammenstellen.

Es ist jedoch kein Werkzeug zum:

Erzeugen,

Reparieren,

Resynchronisieren

oder Rekonfigurieren

eines RAID-Verbunds.

USB

Mit:

```
inxi -J
```

werden USB-Daten ausgegeben.

Dabei können USB-Hubs und angeschlossene USB-Geräte dargestellt werden.

Bedeutung für Rescue

Das ist besonders nützlich bei:

externen Datenträgern,

USB-SATA-Adaptern,

USB-NVMe-Gehäusen,

Netzwerkadaptern,

Tastaturen,

Mäusen

und anderer Rescue-Peripherie.

Bluetooth

Mit:

```
inxi -E
```

können Bluetooth-Informationen ausgegeben werden.

Dazu gehören abhängig von der verfügbaren Systemunterstützung Bluetooth-Geräte, Treiber und Statusinformationen.

Batterie

Mit:

```
inxi -B
```

werden Batterieinformationen ausgegeben, sofern eine Batterie erkannt wird.

Dies kann bei Notebooks im Rescue-Betrieb wichtig sein.

Rescue-Regel für Notebooks

Vor längeren:

Datenrettungen,

Kopiervorgängen,

Image-Erstellungen

oder Datenträgerprüfungen

sollte eine stabile Stromversorgung sichergestellt werden.

Eine Batterieanzeige ersetzt nicht die Sicherstellung einer zuverlässigen Energieversorgung.

Sensoren

Mit:

```
inxi -s
```

können Sensorinformationen ausgegeben werden.

Inxi kann dafür Daten aus verfügbaren Systemquellen beziehungsweise Sensorwerkzeugen zusammenführen.

Abgrenzung zu A.33

Für die eigentliche Sensorabfrage und -konfiguration ist:

A.33 lm-sensors

das spezialisierte Werkzeug.

Inxi integriert Sensorinformationen dagegen in einen Systembericht.

Zusätzliche Details mit `-x`

Inxi besitzt drei zusätzliche Detailstufen:

```
-x
```

```
-xx
```

```
-xxx.
```

Diese ergänzen abhängig vom jeweiligen Informationsbereich weitere Daten.

Beispielprinzip

```
inxi -Cxxx
```

liefert deutlich mehr CPU-Details als:

`inxi -C`.

Analog können die zusätzlichen Detailstufen mit anderen Kategorien kombiniert werden.

Mehr Details können sensible Daten enthalten

Mit höheren Detailstufen können zusätzliche Angaben erscheinen, beispielsweise:

Seriennummern,

Gerätekennungen,

Businformationen

oder weitere technische Identifikatoren.

Vor der Weitergabe eines solchen Reports ist die Filterung deshalb besonders wichtig.

`-a` – administrative Details

Mit:

`-a`

beziehungsweise:

`--admin`

werden zusätzliche administrative Systeminformationen aktiviert.

Die aktuelle Dokumentation beschreibt diese Daten ausdrücklich als erweiterte Informationen für Systemadministratoren.

Bedeutung

Ein administrativer Bericht kann bei komplexen Problemen wertvoll sein.

Er ist jedoch umfangreicher und möglicherweise sensibler als ein normaler Basisbericht.

Nicht automatisch `-a` verwenden

Für die erste Rescue-Inventarisierung genügt meist:

`inxi -b`

oder eine gezielte Kategorie.

`-a` sollte eingesetzt werden, wenn die zusätzlichen technischen Details tatsächlich benötigt werden.

Treiberinformationen

Eine wichtige Stärke von Inxi ist die Verbindung von Hardwareinformationen mit den verwendeten Treibern.

Damit kann beispielsweise sichtbar werden:

welches Netzwerkgerät erkannt wurde

und:

welcher Treiber dafür verwendet wird.

Hardware erkannt, aber falscher Treiber

Gerade bei:

Grafik,

WLAN,

Ethernet,

Audio

oder anderen PCI-/USB-Geräten

kann diese Kombination entscheidend sein.

Das reine Wissen, welches Gerät vorhanden ist, reicht bei einem Treiberproblem nicht aus.

Kernel

Inxi stellt Informationen über den laufenden Kernel bereit.

Im JUMBO BLOCK Rescue Betrieb handelt es sich dabei um den Kernel des gestarteten Rescue-Systems.

Sehr wichtige Abgrenzung

Wird ein beschädigtes installiertes Linux-System mit JUMBO BLOCK Rescue untersucht, beschreibt Inxi grundsätzlich:

das aktuell laufende Rescue-System.

Es beschreibt nicht automatisch den Kernel, die Treiber oder den Desktop des nicht gestarteten Betriebssystems auf der Festplatte.

Beispiel

Wenn das installierte Zielsystem einen alten Kernel besitzt, JUMBO BLOCK Rescue aber mit einem neueren Kernel gestartet wurde, zeigt Inxi den:

Rescue-Kernel.

Das muss bei der Interpretation eines Reports berücksichtigt werden.

Dasselbe gilt für Treiber

Die von Inxi dargestellten aktiven Treiber gehören zur laufenden Rescue-Umgebung.

Sie beweisen nicht, dass das installierte Zielbetriebssystem dieselben Treiber besitzt oder verwendet.

Desktop-Umgebung

Inxi kann auch Informationen zur aktuell laufenden Desktop-/Window-Manager-Umgebung ausgeben.

Auch diese Angaben beziehen sich im Live-Betrieb auf JUMBO BLOCK Rescue.

Betriebssysteminformationen

Entsprechende Systemangaben beziehen sich ebenfalls auf die aktive Umgebung.

Ein Inxi-Bericht aus JUMBO BLOCK Rescue darf daher nicht ohne Kennzeichnung als Bericht des auf der Festplatte installierten Betriebssystems bezeichnet werden.

Rescue-Grundregel

Bei jedem gespeicherten Inxi-Bericht sollte klar dokumentiert werden:

Der Bericht wurde aus der laufenden JUMBO BLOCK Rescue Umgebung erzeugt.

Ausgabe ohne Farben

Für Protokolle und Dateien können Farbcodes unerwünscht sein.

Inxi unterstützt mit:

`-c 0`

die Ausgabe ohne Farbcodes.

Nutzen für gespeicherte Reports

Für einen reinen Textbericht ist eine farblose Ausgabe häufig leichter:

zu speichern,

zu durchsuchen,

zu vergleichen

und weiterzuverarbeiten.

Ausgabe umleiten

Wie bei anderen Kommandozeilenprogrammen kann die Ausgabe über die Shell in eine Datei geschrieben werden.

Beispielprinzip:

```
inxi -b -c 0 > inxi-report.txt
```

Gefilterter Supportbericht

Beispielprinzip:

```
inxi -Fz -c 0 > inxi-support.txt
```

Anschließend sollte die Datei vor Weitergabe kontrolliert werden.

Bericht nicht blind veröffentlichen

Auch ein mit `-z` gefilterter Bericht kann technische Informationen über das System enthalten.

Dazu gehören beispielsweise:

Hardwaremodelle,

Kernelversion,

Treiber,

Speichergrößen

und Systemarchitektur.

Die Filterfunktion reduziert sensible Identifikatoren, macht den Bericht aber nicht informationsfrei.

Ausgabe für interne Dokumentation

Bei rein interner Dokumentation können zusätzliche Details sinnvoll sein.

Dann kann beispielsweise:

`-x`

`-xx`

oder:

`-xxx`

hinzugefügt werden.

Die konkrete Detailtiefe sollte vom Diagnosezweck abhängen.

Typischer Rescue-Ablauf: erster Systemüberblick

1. Basisbericht

```
inxi -b
```

2. Auffälligen Bereich identifizieren

CPU?

RAM?

Storage?

Grafik?

Netzwerk?

3. relevante Kategorie detaillierter untersuchen

beispielsweise:

```
inxi -C
inxi -m
inxi -D
inxi -G
inxi -N
```

4. bei Bedarf Detailstufe erhöhen

```
-x
-xx
-xxx
```

5. spezialisiertes Werkzeug einsetzen

HWInfo,

SMART,

NVMe,

dmidecode,

lm-sensors

oder Netzwerkdiagnose.

Typischer Rescue-Fall: Netzwerk funktioniert nicht

1.

```
inxi -N
```

Netzwerkhardware und Treiber untersuchen.

2.

erweiterte Netzwerkinformationen prüfen.

3.

danach mit spezialisierten Netzwerkwerkzeugen fortfahren.

Beispielsweise:

Dig,

MTR,

Nmap,

ARP Scan,

TCPDump

oder Wireshark.

Typischer Rescue-Fall: Grafikproblem

1.

`inxi -G`

2.

bei Bedarf:

`inxi -Gx`

beziehungsweise höhere Detailstufe.

3.

Hardware und aktiven Treiber vergleichen.

4.

gegebenenfalls mit HWInfo ergänzen.

Typischer Rescue-Fall: Storage

1.

`inxi -D`

2.

Partitionen und Mountzustände untersuchen.

3.

mit HWInfo die Geräteidentität ergänzen.

4.

SMART beziehungsweise NVMe-Gesundheitsdaten prüfen.

5.

bei physischem Schadensverdacht keine unnötige Belastung erzeugen.

6.

gegebenenfalls mit GNU ddrescue zuerst ein Abbild erstellen.

Typischer Rescue-Fall: Hardwarebericht für Support

1. benötigte Detailtiefe festlegen.

2. Filterung sensibler Angaben aktivieren.

3. Farbausgabe für Textdatei deaktivieren.

4. Bericht speichern.

5. Bericht manuell auf sensible Angaben kontrollieren.

6. erst danach weitergeben.

Typischer Rescue-Fall: Notebook

Ein Bericht kann insbesondere Informationen zu:

CPU,

Grafik,

Netzwerk,

Batterie,

Storage

und Sensoren

zusammenführen.

Dadurch entsteht schnell eine brauchbare technische Übersicht des aktuell laufenden Systems.

Inxi und HWInfo

HWInfo

liefert detaillierte Hardwareerkennung nach Geräteklassen.

Inxi

verdichtet viele dieser Informationen zu einem lesbaren Gesamtbericht und ergänzt System-, Treiber- und Laufzeitinformationen.

Für schwierige Fälle können beide Ergebnisse miteinander verglichen werden.

Inxi und LSHW

LSHW

stellt die Hardware hierarchisch dar.

Inxi

stellt die Informationen stärker nach funktionalen Kategorien und als Systembericht zusammen.

Inxi und dmidecode

dmidecode

liefert gezielt SMBIOS-/DMI-Daten.

Inxi kann Teile solcher Systeminformationen in einen größeren Bericht einbeziehen.

Für die genaue DMI-Untersuchung bleibt dmidecode das spezialisierte Werkzeug.

Inxi und lm-sensors

Inxi kann Sensorwerte anzeigen.

`lm-sensors` ist jedoch das eigentliche Sensorwerkzeug.

Damit lautet die Rollenverteilung:

Inxi

Sensorinformationen im Gesamtbericht.

lm-sensors

gezielte Sensorabfrage und Sensorkontext.

Inxi und SMART/NVMe

Inxi kann Datenträgerdaten und – abhängig von Optionen und verfügbaren Werkzeugen – zusätzliche Storage-Informationen darstellen.

Für eine belastbare spezialisierte Zustandsdiagnose bleiben jedoch:

`smartctl`

und:

`nvme-cli`

die maßgeblichen Werkzeuge dieses Rescue-Handbuchs.

Externe Hilfsprogramme

Inxi sammelt Informationen aus unterschiedlichen Systemquellen und kann für bestimmte Funktionen auf vorhandene Hilfsprogramme zurückgreifen.

Deshalb kann die tatsächlich verfügbare Ausgabe abhängig sein von:

Betriebssystem,

Hardware,

Kernel,

Berechtigungen

und installierten Hilfswerkzeugen.

Root-Rechte

Bestimmte tiefere Informationen können erhöhte Rechte oder privilegierte Datenquellen erfordern.

Nicht jede Inxi-Ausgabe benötigt jedoch Root-Rechte.

Rescue-Regel

Nicht pauschal mit Root starten.

Zunächst normalen Bericht erzeugen.

Erhöhte Rechte nur einsetzen, wenn die benötigte Information dies tatsächlich erfordert.

Grenzen

Inxi ist ein Informations- und Diagnosewerkzeug.

Inxi repariert keine Hardware.

Inxi repariert keine Dateisysteme.

Inxi repariert keine Partitionstabellen.

Inxi führt keinen vollständigen RAM-Test durch.

Inxi führt keinen CPU-Stresstest durch.

Inxi ersetzt SMART nicht.

Inxi ersetzt `nvme-cli` nicht.

Inxi ersetzt `lm-sensors` nicht.

Inxi ersetzt `dmidecode` für eine gezielte SMBIOS-/DMI-Analyse nicht.

Inxi kann nur Informationen darstellen, die der laufenden Umgebung zugänglich sind.

Ein Live-Inxi-Bericht beschreibt primär die laufende Rescue-Umgebung und deren Sicht auf die Hardware.

Sicherheitshinweise

Inxi-Berichte vor externer Weitergabe immer prüfen.

Für Supportberichte sensible Informationen mit `-z` filtern.

IP-Adressen, MAC-Adressen, Seriennummern, Hostnamen und Benutzerinformationen als potenziell sensibel behandeln.

Hohe Detailstufen nur verwenden, wenn sie benötigt werden.

Nicht aus einer erkannten Hardwarekomponente automatisch auf deren Fehlerfreiheit schließen.

Bei Storage-Problemen Inxi nur zur Identifikation und Übersicht verwenden; Gesundheitsdiagnose mit den spezialisierten Werkzeugen durchführen.

Bei physisch instabilen Datenträgern keine unnötigen Abfragen oder Belastungen erzeugen.

Bei Live-Systemen stets dokumentieren, dass Kernel, Treiber und Desktopinformationen zur Rescue-Umgebung gehören.

Technische Einordnung in JUMBO BLOCK Rescue

Inxi ist das zweite Werkzeug des Bereichs:

Hardware.

Die Rollenverteilung lautet:

A.29 HWInfo

gezielte Hardwareerkennung nach Geräteklassen.

↓

A.30 Inxi

konsolidierter System- und Hardwarebericht.

↓

A.31 LSHW

hierarchische Hardwarestruktur.

↓

A.32 DMI Decode

SMBIOS-/DMI-Daten.

↓

A.33 Sensoren

laufende Sensorinformationen.

Ein sinnvoller Rescue-Ablauf kann deshalb beginnen mit:

```
inxi -b
```

für den schnellen Gesamtüberblick.

Danach:

HWInfo

für eine bestimmte Hardwareklasse.

LSHW

für die hierarchische Gerätezuordnung.

dmidecode

für Firmware-/DMI-Informationen.

lm-sensors

für Temperatur- und andere unterstützte Sensorwerte.

Bei Storage-Problemen folgen:

SMART / NVMe

und gegebenenfalls:

GNU ddrescue.

Die besondere Stärke von Inxi liegt damit nicht in einer einzelnen Hardwareprüfung, sondern in der Zusammenführung vieler Informationsquellen zu einem schnell lesbaren technischen Gesamtbild.

Für JUMBO BLOCK Rescue lautet die Kernfrage von Inxi:

Wie sieht das aktuell laufende Rescue-System mit der von ihm erkannten Hardware, seinen Treibern und seinen wesentlichen Systemparametern als Gesamtbild aus?

A.31 LSHW

Bereich: Hardware
Programm: LSHW – List Hardware
Baseline 20261001: 02.19.git.2021.06.19.996aaad9c7
Installiertes Paket: `lshw 02.19.git.2021.06.19.996aaad9c7-2.1`
Bedienung: Kommandozeile
Aufgabe: Detaillierte und hierarchische Inventarisierung der vom laufenden System erkannten Hardware

Zweck

LSHW steht für:

List Hardware.

Das Programm extrahiert detaillierte Informationen über die Hardwarekonfiguration eines Rechners und ordnet erkannte Komponenten in einer hierarchischen Gerätebaumstruktur an.

Die exakt zur installierten Baseline gehörende Debian-Manpage nennt unter anderem Informationen über:

Speicherkonfiguration,

Firmware-Version,

Mainboard-Konfiguration,

CPU-Version und -Geschwindigkeit,

Cache-Konfiguration,

Busgeschwindigkeiten

und weitere Hardwareeigenschaften.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue dient LSHW insbesondere dazu:

die Hardwarestruktur eines unbekannten Systems zu inventarisieren,

Beziehungen zwischen Geräten und übergeordneten Komponenten zu erkennen,

Storage-Controller und Datenträger gemeinsam zu untersuchen,

Netzwerkhardware und deren Konfiguration zu erfassen,

CPU und Speicherstruktur zu dokumentieren,

PCI-, USB- und SCSI-Geräte zuzuordnen,

Busadressen zu ermitteln,

Treiber- und Gerätekonfigurationen zu untersuchen

und einen Hardwarebericht für Diagnose oder Dokumentation zu erzeugen.

Unterschied zu HWInfo und Inxi

Die drei bisherigen Hardwarewerkzeuge besitzen unterschiedliche Schwerpunkte.

HWInfo

ermöglicht gezielte Hardwareerkennung nach Geräteklassen.

Inxi

erzeugt einen konsolidierten und gut lesbaren Systembericht.

LSHW

stellt Hardware besonders als hierarchischen Gerätebaum dar.

Die zentrale Frage von LSHW lautet damit:

Wie sind die vom System erkannten Hardwarekomponenten strukturell miteinander verbunden und konfiguriert?

Grundaufruf

Der grundlegende Aufruf lautet:

```
lshw
```

LSHW erzeugt damit einen detaillierten Hardwarebaum.

Root-Rechte

Die zur Baseline gehörende Manpage weist ausdrücklich darauf hin, dass LSHW als Superuser ausgeführt werden muss, wenn vollständige Informationen gewünscht werden.

Ohne entsprechende Rechte wird nur ein Teil der verfügbaren Informationen ausgegeben.

Rescue-Regel

Ein zunächst unprivilegiert erzeugter Bericht kann für einen schnellen Überblick ausreichen.

Wird eine vollständige Hardwareinventarisierung benötigt, muss berücksichtigt werden, dass fehlende Informationen lediglich auf eingeschränkte Zugriffsrechte zurückzuführen sein können.

Hierarchische Darstellung

Die normale LSHW-Ausgabe ist hierarchisch aufgebaut.

Dadurch lassen sich Komponenten nicht nur einzeln betrachten, sondern auch in ihrem Hardwarekontext.

Beispielsweise kann eine Struktur sinngemäß enthalten:

System

→ Bus

→ Storage-Controller

→ Datenträger

oder:

System

→ Bus

→ Netzwerkcontroller

→ Netzwerkschnittstelle.

Bedeutung im Rescue-Betrieb

Diese Hierarchie kann entscheidend sein, wenn beispielsweise mehrere:

Storage-Controller,

Netzwerkadapter,

USB-Geräte

oder Datenträger

vorhanden sind.

Die Frage lautet dann nicht nur:

Welches Gerät existiert?

sondern:

Über welchen Bus beziehungsweise Controller ist dieses Gerät eingebunden?

Kurzansicht

Mit:

```
lshw -short
```

wird der Hardwarebaum in einer kompakten Form ausgegeben.

Die Manpage beschreibt diese Darstellung anhand der Hardwarepfade und nennt sie ausdrücklich als kompakte Hardwareliste.

Typischer Einstieg

Für eine erste Übersicht ist deshalb:

```
lshw -short
```

besonders geeignet.

Danach können interessante Hardwareklassen gezielt untersucht werden.

Hardwarepfad

Die Kurzansicht zeigt Hardwarepfade.

Diese Pfade helfen dabei, die hierarchische Position eines Geräts innerhalb des von LSHW ermittelten Hardwarebaums zu erkennen.

Sie sind nicht mit normalen Dateisystempfaden wie:

`/home/...`

oder:

`/dev/...`

gleichzusetzen.

Businformationen

Mit:

`lshw -businfo`

wird eine Geräteliste mit Businformationen erzeugt.

Die Manpage nennt dabei insbesondere Adressen für:

SCSI,

USB,

IDE

und PCI.

Bedeutung

Damit kann beispielsweise untersucht werden, welchem:

PCI-Gerät,

USB-Gerät

oder SCSI-Pfad

eine erkannte Hardwarekomponente zugeordnet ist.

`-short` und `-businfo`

Beide Darstellungen dienen einer kompakten Übersicht, setzen aber unterschiedliche Schwerpunkte:

`-short`

→ Hardwarepfade.

`-businfo`

→ Buszuordnungen.

Hardwareklassen

Mit:

```
-class CLASS
```

kann die Ausgabe auf eine bestimmte Hardwareklasse begrenzt werden.

Die Kurzform lautet:

`-C CLASS`.

Klassen ermitteln

Die verfügbaren beziehungsweise im konkreten System vorkommenden Klassen können anhand von:

```
lshw -short
```

oder:

```
lshw -businfo
```

ermittelt werden.

Mehrere Klassen kombinieren

```
-class
```

kann mehrfach angegeben werden.

Die Manpage nennt als Beispiel:

```
lshw -class disk -class storage
```

Damit werden Datenträger und Storage-Controller gemeinsam angezeigt.

Storage-Diagnose

Gerade diese Kombination ist für JUMBO BLOCK Rescue besonders sinnvoll:

```
lshw -class disk -class storage
```

Sie beantwortet zwei zusammengehörige Fragen:

Welche Datenträger erkennt das System?

und:

Welche Storage-Controller erkennt das System?

Netzwerkhardware

Eine gezielte Abfrage kann beispielsweise erfolgen mit:

```
lshw -class network
```

Damit wird die Ausgabe auf Netzwerkhardware eingeschränkt.

Netzwerkbericht als HTML

Die zur Baseline gehörende Manpage nennt ausdrücklich:

```
lshw -html -class network
```

als Beispiel zur Ausgabe der Netzwerkhardware in HTML.

Konfiguration eines Netzwerkgeräts

LSHW kann bei Netzwerkhardware nicht nur die Geräteidentität, sondern abhängig von den verfügbaren Informationen auch Konfigurationsdaten anzeigen.

Dazu können beispielsweise Angaben zur verwendeten Schnittstelle und Treiberzuordnung gehören.

Hardware erkannt bedeutet nicht Netzwerk funktioniert

Wie bei HWInfo und Inxi gilt:

Ein erkannter Netzwerkcontroller beweist nicht, dass:

eine korrekte IP-Konfiguration vorhanden ist,

DNS funktioniert,

das Gateway erreichbar ist

oder eine Netzwerkverbindung erfolgreich aufgebaut werden kann.

Dafür stehen die Netzwerkwerkzeuge aus A.17 bis A.24 zur Verfügung.

CPU

LSHW kann detaillierte Prozessorinformationen erfassen.

Die Manpage nennt ausdrücklich:

CPU-Version

und:

CPU-Geschwindigkeit

als Beispiele der ermittelbaren Hardwaredaten.

CPU-Erkennung ist kein Belastungstest

LSHW inventarisiert die CPU.

Es testet jedoch nicht die Stabilität des Prozessors unter Last.

Eine korrekt erkannte CPU ist deshalb nicht automatisch nachweislich fehlerfrei.

Cache

LSHW kann auch die Cache-Konfiguration erfassen.

Dadurch kann die CPU-Hierarchie zusätzliche Cache-Komponenten enthalten.

Arbeitsspeicher

LSHW kann detaillierte Informationen über die Speicherkonfiguration ausgeben.

Auf unterstützten Systemen können dabei verschiedene Speicherstrukturen und Speicherbänke sichtbar werden.

SPD-Erkennung

LSHW besitzt einen Hardwaretest:

`spd`

für:

Serial Presence Detect

von Speichermodulen.

Dieser Test kann über die allgemeinen Mechanismen:

`-enable`

beziehungsweise:

`-disable`

gesteuert werden.

RAM-Inventarisierung ist kein RAM-Test

Auch detaillierte Angaben über Speicherbänke oder Module beweisen nicht, dass der Arbeitsspeicher fehlerfrei arbeitet.

LSHW führt keinen vollständigen Speicherfehlertest durch.

Firmware und Mainboard

LSHW kann Firmware- und Mainboardinformationen erfassen.

Die Manpage nennt:

Firmware-Version

und:

Mainboard-Konfiguration

ausdrücklich als typische Ausgaben.

DMI

Auf x86- und IA-64-Systemen unterstützt LSHW:

DMI beziehungsweise SMBIOS.

Diese Informationen können einen Teil der Hardwareinventarisierung bilden.

Abgrenzung zu dmidecode

Das folgende Kapitel:

A.32 DMI Decode

behandelt mit `dmidecode` ein spezialisiertes Werkzeug für SMBIOS-/DMI-Strukturen.

LSHW verwendet DMI dagegen als eine von mehreren Informationsquellen und integriert die Ergebnisse in seinen Hardwarebaum.

Unterstützte Informationsquellen

Die exakt zur Baseline gehörende Manpage nennt Unterstützung für:

DMI,

OpenFirmware Device Tree,

PCI/AGP,

CPUID,

IDE/ATA/ATAPI,

PCMCIA,

SCSI

und USB.

Linux-Systeminformationen

LSHW greift zusätzlich auf verschiedene Kernel- und Systeminformationen zu.

Die Manpage nennt unter anderem:

`/sys/*`

sowie verschiedene `/proc`- und Gerätepfade als Informationsquellen.

Rescue-Perspektive

Damit beschreibt LSHW die Hardware aus Sicht des aktuell laufenden:

JUMBO BLOCK Rescue Systems.

Wie bereits bei Inxi gilt:

Treiber- und Laufzeitinformationen beziehen sich auf die Rescue-Umgebung und nicht automatisch auf ein Betriebssystem, das auf einem untersuchten Datenträger installiert ist.

PCI und USB

LSHW kann PCI- und USB-Geräte erkennen und in den Hardwarebaum einordnen.

Für eine kompakte Zuordnung eignet sich insbesondere:

```
lshw -businfo
```

Numerische Gerätekennungen

Mit:

```
-numeric
```

werden zusätzlich numerische IDs für PCI- und USB-Geräte angezeigt.

Nutzen numerischer IDs

Hersteller- und Gerätekennungen können hilfreich sein, wenn:

ein Gerät nur generisch erkannt wird,

ein Treiberproblem untersucht wird

oder die genaue PCI-/USB-Geräteidentität benötigt wird.

Beispielprinzip

```
lshw -numeric -class network
```

kann Netzwerkhardware zusammen mit zusätzlichen numerischen Gerätekennungen darstellen.

Datenträger

Die Klasse:

```
disk
```

dient zur gezielten Anzeige erkannter Datenträger.

Storage-Controller

Die Klasse:

```
storage
```

dient zur Anzeige entsprechender Storage-Hardware.

Kombination

Für einen Rescue-Fall ist daher besonders sinnvoll:

```
lshw -class disk -class storage
```

Diese Kombination ist sogar als Beispiel in der offiziellen Manpage der installierten Version dokumentiert.

Datenträger erkannt bedeutet nicht Datenträger gesund

LSHW kann Modell, Gerätezuordnung und andere Hardwaredaten liefern.

Es ist jedoch kein Werkzeug zur Analyse von:

SMART-Attributen,

NVMe-Health-Daten,

Medienfehlern

oder physischem Verschleiß.

Zustandsdiagnose

Hierfür stehen in JUMBO BLOCK Rescue bereit:

A.8 SMART Diagnose

und:

A.9 NVMe Diagnose.

Hardwarebaum und Partitionen

LSHW kann Storage-Strukturen im Hardwarebaum darstellen.

Für eine gezielte Partitionsbearbeitung ist jedoch:

A.6 GParted

zuständig.

Für die grafische Datenträgerverwaltung:

A.7 GNOME Disks.

HTML-Ausgabe

Mit:

`-html`

wird der Gerätebaum als HTML-Seite ausgegeben.

Dies kann für eine lesbare Hardwaredokumentation nützlich sein.

XML-Ausgabe

Mit:

`-xml`

wird der Gerätebaum als XML-Struktur ausgegeben.

Dies eignet sich insbesondere für eine maschinelle Weiterverarbeitung.

JSON-Ausgabe

Mit:

`-json`

wird der Gerätebaum als JSON ausgegeben.

Damit kann die Hardwarestruktur von Skripten oder anderen Werkzeugen weiterverarbeitet werden.

Ausgabeformat bewusst wählen

Für unterschiedliche Rescue-Zwecke bieten sich damit unterschiedliche Formate an:

normale Ausgabe

menschliche Detailanalyse.

-short

schnelle Übersicht.

-businfo

Buszuordnung.

HTML

lesbarer Bericht.

XML / JSON

strukturierte Weiterverarbeitung.

Bericht in Datei speichern

Wie bei anderen Kommandozeilenprogrammen kann die normale Ausgabe über die Shell in eine Datei umgeleitet werden.

Beispiel:

```
lshw -short > lshw-short.txt
```

HTML-Bericht

Beispiel:

```
lshw -html > lshw-report.html
```

JSON-Bericht

Beispiel:

```
lshw -json > lshw-report.json
```

Sensible Informationen

Hardwareberichte können identifizierende Daten enthalten.

Dazu können beispielsweise gehören:

Seriennummern,

IP-Adressen,

Hardwarekennungen

und weitere systemspezifische Informationen.

`-sanitize`

LSHW besitzt deshalb die Option:

`-sanitize`

Sie entfernt potenziell sensible Informationen aus der Ausgabe.

Die Manpage nennt ausdrücklich:

IP-Adressen,

Seriennummern

und ähnliche Daten.

Supportbericht

Für einen extern weiterzugebenden Bericht kann deshalb beispielsweise eine Kombination mit:

`-sanitize`

sinnvoll sein.

Beispielprinzip

```
lshw -sanitize -short
```

oder:

```
lshw -sanitize -html > hardware.html
```

Filterung trotzdem kontrollieren

Auch ein sanitizierter Hardwarebericht enthält weiterhin technische Informationen über das System.

Vor einer Veröffentlichung sollte der Bericht deshalb zusätzlich manuell geprüft werden.

`-notime`

Mit:

`-notime`

werden volatile zeitbezogene Attribute wie Zeitstempel aus der Ausgabe ausgeschlossen.

Nutzen für Vergleichsberichte

Dies kann hilfreich sein, wenn zwei Hardwareberichte automatisiert miteinander verglichen werden sollen und flüchtige Zeitinformationen den Vergleich unnötig verändern würden.

`-quiet`

Mit:

`-quiet`

werden Statusmeldungen unterdrückt.

Dies kann insbesondere bei:

Dateiausgabe,

Skripten

oder automatisierter Verarbeitung

nützlich sein.

Hardwaretests aktivieren oder deaktivieren

LSHW unterstützt:

`-enable TEST`

und:

`-disable TEST`.

Damit können bestimmte Erkennungsmechanismen ein- beziehungsweise ausgeschaltet werden.

Dokumentierte Tests

Die Baseline-Manpage nennt:

`dmi`

`device-tree`

`spd`

`memory`

`cpuinfo`

`cpuid`

`pci`

`isapnp`

`pcmcia`

`ide`

`usb`

`scsi`

und:

`network`.

Bedeutung von „Test“

Der Begriff darf hier nicht mit einem Hardware-Stresstest verwechselt werden.

Diese Tests bezeichnen Erkennungsmechanismen beziehungsweise Informationsquellen von LSHW.

Beispiel

Die Manpage nennt:

```
lshw -disable dmi
```

als Beispiel.

Damit wird DMI für die Hardwareerkennung nicht verwendet.

Warum Informationsquellen deaktivieren?

Dies kann beispielsweise bei:

Fehlersuche,

Vergleich unterschiedlicher Erkennungswege

oder problematischen Informationsquellen

nützlich sein.

Für einen normalen Rescue-Bericht ist es jedoch nicht erforderlich, Erkennungsmethoden ohne konkreten Grund abzuschalten.

SQLite-Dump

LSHW unterstützt:

`-dump DATEINAME`.

Damit können die gesammelten Informationen in eine SQLite-Datenbank geschrieben werden.

Bedeutung

Dies ermöglicht eine strukturierte Speicherung der Hardwareinformationen für spätere Auswertung.

Für einen normalen menschlich lesbaren Rescue-Bericht sind jedoch:

Text,

HTML

oder:

JSON

häufig direkter verwendbar.

Versionsabfrage

Mit:

```
lshw -version
```

wird die LSHW-Version ausgegeben.

Hilfe

Mit:

```
lshw -help
```

werden die verfügbaren Kommandozeilenoptionen angezeigt.

Grafische Oberfläche

LSHW kennt:

```
-X
```

zum Start einer X11-GUI, sofern diese verfügbar ist.

Baseline-Abgrenzung zur GUI

Die Paketbaseline belegt:

```
lshw
```

in Version:

`02.19.git.2021.06.19.996aaad9c7-2.1`.

Aus dieser Paketzeile allein wird für JUMBO BLOCK Rescue keine zusätzlich installierte `lshw-gtk`-Oberfläche abgeleitet.

Für dieses Kapitel wird deshalb ausschließlich die belegte Kommandozeilenfunktion von `lshw` vorausgesetzt.

Typischer Rescue-Fall: unbekannter Rechner

Ein sinnvoller Ablauf lautet:

1. Kompakte Hardwareübersicht

```
lshw -short
```

2. Buszuordnung

```
lshw -businfo
```

3. auffällige Hardwareklasse identifizieren

4. diese Klasse gezielt untersuchen

beispielsweise:

```
lshw -class network
```

oder:

```
lshw -class disk -class storage
```

5. bei Bedarf mit HWInfo, Inxi oder einem Spezialwerkzeug ergänzen

Typischer Rescue-Fall: Datenträger fehlt

1.

```
lshw -class disk -class storage
```

2.

Buszuordnung mit:

```
lshw -businfo
```

prüfen.

3.

Storage-Controller identifizieren.

4.

gegebenenfalls mit HWInfo ergänzen.

5.

wenn der Datenträger sichtbar wird, SMART- beziehungsweise NVMe-Diagnose durchführen.

Typischer Rescue-Fall: Netzwerkadapter funktioniert nicht

1.

```
lshw -class network
```

2.

Gerät, Buszuordnung und verfügbare Konfigurationsinformationen prüfen.

3.

bei Bedarf:

```
lshw -numeric -class network
```

verwenden.

4.

mit HWInfo oder Inxi gegenprüfen.

5.

anschließend Netzwerkfunktion mit den Netzwerkwerkzeugen untersuchen.

Typischer Rescue-Fall: USB-Gerät unbekannt

1.

`lshw -businfo`

2.

USB-Buszuordnung identifizieren.

3.

bei Bedarf numerische Gerätekennungen ergänzen.

4.

mit HWInfo `--usb` gegenprüfen.

Typischer Rescue-Fall: Hardwaredokumentation

Für eine weiterzugebende Dokumentation kann beispielsweise:

`lshw -sanitize -html`

verwendet und die Ausgabe in eine HTML-Datei geschrieben werden.

Vor Weitergabe:

Bericht manuell auf sensible Informationen kontrollieren.

Typischer Rescue-Fall: maschineller Vergleich

Für eine strukturierte Weiterverarbeitung können:

JSON

oder:

XML

verwendet werden.

Mit:

`-notime`

lassen sich zeitabhängige Attribute ausblenden, wenn sie für einen Vergleich störend sind.

HWInfo, Inxi und LSHW gemeinsam

Die drei Werkzeuge können bei komplexen Fällen bewusst miteinander kombiniert werden.

HWInfo

fragt gezielt Geräteklassen ab.

Inxi

liefert einen kompakten Gesamtbericht aus Hardware- und Systeminformationen.

LSHW

zeigt die hierarchische Geräte- und Busstruktur.

Wenn alle drei Werkzeuge dieselbe Hardware erkennen, entsteht eine gute technische Grundlage für die weitere Diagnose.

Abweichende Ergebnisse

Unterschiede zwischen den Werkzeugen bedeuten nicht automatisch, dass eines davon fehlerhaft arbeitet.

Sie können unterschiedliche:

Informationsquellen,

Darstellungsmodelle,

Zugriffsrechte

und Erkennungsverfahren

verwenden.

Die Ursache einer Abweichung muss deshalb untersucht werden.

Grenzen der Hardwareerkennung

Die Baseline-Manpage nennt ausdrücklich Einschränkungen.

LSHW erkennt dort unter anderem FireWire-/IEEE-1394-Geräte nicht.

Außerdem werden nicht alle von GNU/Linux unterstützten Architekturen vollständig unterstützt; als Beispiel nennt die Manpage die CPU-Erkennung.

Auch virtuelle SCSI-Schnittstellen zur SCSI-Emulation über IDE werden laut Dokumentation nicht korrekt dargestellt.

Wichtige Schlussfolgerung

„Nicht in LSHW vorhanden“ bedeutet nicht automatisch „physisch nicht vorhanden“.

Die Ausgabe beschreibt, was LSHW mit seinen unterstützten Erkennungsmechanismen feststellen konnte.

Hardwaregesundheit

LSHW führt keine vollständige Gesundheitsprüfung der Komponenten durch.

Es testet nicht:

Datenträgeroberflächen,

SMART-Zustand,

NVMe-Health,

RAM-Zellen,

CPU-Stabilität

oder elektrische Sensorgrenzen.

Schreibwirkung

Die üblichen Inventarisierungsaufrufe dienen dem Auslesen und Darstellen von Hardwareinformationen.

LSHW ist kein Werkzeug zur Partitionierung oder Dateisystemreparatur.

Trotzdem mit Bedacht einsetzen

Auf instabiler oder fehlerhafter Hardware können auch Diagnosezugriffe Auswirkungen haben.

Bei einem physisch sterbenden Datenträger sollte deshalb nicht unnötig eine große Zahl verschiedener Diagnoseprogramme wiederholt darauf angesetzt werden.

Wenn die Datenrettung Priorität besitzt, kann eine möglichst frühe Acquisition mit GNU ddrescue wichtiger sein.

Live-System-Perspektive

Wie HWInfo und Inxi arbeitet LSHW aus der laufenden Rescue-Umgebung.

Die erkannten Geräte und aktiven Konfigurationen werden damit aus Sicht des:

JUMBO BLOCK Rescue Kernels

ermittelt.

Installiertes Zielsystem

Ein LSHW-Bericht aus JUMBO BLOCK Rescue ist deshalb nicht automatisch ein Bericht über die Treiberkonfiguration eines beschädigten Linux-Systems auf der Festplatte.

Er dokumentiert die aktuelle Rescue-Sicht auf die Hardware.

Sicherheitshinweise

Für vollständige Hardwareinformationen erforderliche Root-Rechte bewusst einsetzen.

Hardwareberichte vor Weitergabe mit `-sanitize` behandeln und zusätzlich kontrollieren.

Seriennummern, IP-Adressen und Gerätekennungen als potenziell sensible Daten betrachten.

Vor schreibenden Datenträgeroperationen Geräteidentität und Buszuordnung eindeutig prüfen.

Datenträgererkennung nicht mit Datenträgergesundheit verwechseln.

Bei physisch auffälligen Datenträgern unnötige Diagnosezugriffe vermeiden.

Ein fehlendes Gerät in LSHW nicht automatisch als Beweis für physische Abwesenheit interpretieren.

Live-System-Berichte als Sicht von JUMBO BLOCK Rescue kennzeichnen.

Technische Einordnung in JUMBO BLOCK Rescue

LSHW ist das dritte Programm des Hardware-Blocks.

Die Rollenverteilung lautet:

A.29 HWInfo

breite Hardwareerkennung nach Geräteklassen.

↓

A.30 Inxi

konsolidierter Systembericht.

↓

A.31 LSHW

hierarchische Hardware- und Busstruktur.

↓

A.32 DMI Decode

gezielte SMBIOS-/DMI-Informationen.

↓

A.33 Sensoren

aktuelle unterstützte Sensorwerte.

LSHW ergänzt die beiden vorherigen Programme besonders dort, wo nicht nur die Identität eines Geräts, sondern dessen:

Position und Beziehung innerhalb der Hardwarestruktur

relevant ist.

Ein typischer Storage-Diagnosepfad lautet beispielsweise:

```
lshw -short
```

↓

```
lshw -class disk -class storage
```

↓

```
lshw -businfo
```

↓

HWInfo

weitere Geräteerkennung.

↓

SMART / NVMe

Gesundheitszustand.

↓

GNU ddrescue

bei instabilem Quellmedium.

Die zentrale Rolle von LSHW in JUMBO BLOCK Rescue lautet damit:

Hardware nicht nur auflisten, sondern ihre technische Struktur und Zuordnung sichtbar machen.

Gerade vor kritischen Rescue-Arbeiten ist das entscheidend:

Welches Gerät ist vorhanden – und wo hängt es im System?

A.32 DMI Decode

Bereich: Hardware
Programm: dmidecode
Baseline 20261001: dmidecode 3.6
Installiertes Paket: `dmidecode 3.6-2`
Bedienung: Kommandozeile
Aufgabe: SMBIOS-/DMI-Tabellen auslesen und in menschenlesbarer Form darstellen

Zweck

`dmidecode` liest die DMI-/SMBIOS-Tabellen eines Computers aus und decodiert deren Inhalt in eine menschenlesbare Darstellung.

SMBIOS steht für:

System Management BIOS

DMI für:

Desktop Management Interface.

Die Tabellen enthalten von BIOS beziehungsweise UEFI/Firmware bereitgestellte Beschreibungen der Systemhardware.

Dazu gehören typischerweise Informationen über:

Systemhersteller,

Systemmodell,

Seriennummer,

System-UUID,

BIOS beziehungsweise Firmware,

Mainboard,

Gehäuse,

Prozessor,

Cache,

Speicherbestückung,

Steckplätze,

Anschlüsse

und weitere Plattforminformationen.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich dmidecode insbesondere zur:

Identifikation eines unbekannten Rechners,

Ermittlung von Hersteller und Systemmodell,

Dokumentation von BIOS-/Firmwareinformationen,

Identifikation des Mainboards,

Ermittlung von Gehäuseinformationen,

Untersuchung der vom SMBIOS beschriebenen CPU-Konfiguration,

Erfassung von Speicherbänken und Speichermodulen,

Ermittlung von Seriennummern und Asset-Tags,

Anzeige von System-UUIDs,

Untersuchung vorhandener Steckplätze

und Dokumentation der firmwareseitig gemeldeten Hardwarekonfiguration.

Entscheidender Unterschied zu HWInfo und LSHW

dmidecode untersucht die Hardware nicht primär durch aktive Hardwareerkennung.

Stattdessen liest es die vom System bereitgestellte:

SMBIOS-/DMI-Tabelle.

Das hat einen wichtigen Vorteil:

Die Informationen können schnell ausgelesen werden, ohne die tatsächliche Hardware einzeln abfragen zu müssen.

Es hat aber auch eine entscheidende Grenze:

Die Firmwaredaten können falsch, unvollständig oder unzuverlässig sein.

Die dmidecode-Dokumentation weist ausdrücklich auf diesen Umstand hin.

Grundregel

dmidecode zeigt, was die Firmware über die Hardware behauptet – nicht zwingend, was physisch tatsächlich vorhanden ist.

Grundaufruf

Der vollständige Aufruf lautet:

```
dmidecode
```

Damit versucht das Programm, die DMI-Tabelle zu finden, auszulesen und sämtliche unterstützten Einträge zu decodieren.

Root-Rechte

Der vollständige Zugriff auf die SMBIOS-/DMI-Daten erfordert auf typischen Linux-Systemen privilegierten Zugriff.

Ein üblicher Rescue-Aufruf lautet deshalb:

```
sudo dmidecode
```

Datenquelle unter Linux

dmidecode versucht unter Linux zunächst, die DMI-Tabelle über:

sysfs

zu lesen.

Schlägt dieser Zugriff fehl, versucht es anschließend den direkten Speicherzugriff.

Direkter Speicherzugriff

Für den direkten Zugriff verwendet dmidecode standardmäßig:

```
/dev/mem
```

Mit:

```
-d FILE
```

beziehungsweise:

```
--dev-mem FILE
```

kann eine andere Gerätedatei angegeben werden.

Für einen normalen Rescue-Einsatz ist eine Änderung dieser Quelle in der Regel nicht erforderlich.

Aufbau eines DMI-Eintrags

Ein DMI-Datensatz besitzt unter anderem:

einen Handle,

einen Typ,

eine Größe

und:

decodierte Werte.

Ein Handle ist eine eindeutige Kennung, über die DMI-Einträge aufeinander verweisen können.

Beispielsweise kann ein Prozessor-Eintrag auf einen Cache-Eintrag verweisen.

DMI-Typen

SMBIOS definiert verschiedene Datensatztypen.

Zu den für Rescue-Arbeiten besonders wichtigen gehören:

`0` – BIOS

`1` – System

`2` – Baseboard

`3` – Chassis

`4` – Processor

`7` – Cache

`9` – System Slots

`16` – Physical Memory Array

`17` – Memory Device

`22` – Portable Battery

`32` – System Boot

`38` – IPMI Device.

Gezielte Abfrage mit `--type`

Mit:

`-t TYPE`

beziehungsweise:

`--type TYPE`

kann die Ausgabe auf bestimmte DMI-Typen beschränkt werden.

TYPE kann dabei:

eine Typnummer,

eine kommaseparierte Liste von Typnummern

oder ein unterstütztes Schlüsselwort

sein.

Unterstützte Typ-Schlüsselwörter

Version 3.6 unterstützt hierfür die Schlüsselwörter:

`bios`

`system`

`baseboard`

`chassis`

`processor`

`memory`

`cache`

`connector`

`slot`.

BIOS

Für BIOS-/Firmwareinformationen:

```
dmidecode --type bios
```

Das Schlüsselwort umfasst die entsprechenden BIOS-bezogenen DMI-Typen.

System

Für Systeminformationen:

```
dmidecode --type system
```

Hier können beispielsweise Hersteller, Produktbezeichnung, Version, Seriennummer oder UUID erscheinen, soweit die Firmware diese Daten bereitstellt.

Mainboard

Für Mainboardinformationen:

```
dmidecode --type baseboard
```

Damit können beispielsweise Hersteller, Produktname, Version und Seriennummer des Baseboards ermittelt werden.

Gehäuse

Für Gehäuseinformationen:

```
dmidecode --type chassis
```

Prozessor

Für Prozessorinformationen:

```
dmidecode --type processor
```

Speicher

Besonders wichtig für Rescue-Diagnosen ist:

```
dmidecode --type memory
```

Das Schlüsselwort fasst mehrere speicherbezogene DMI-Typen zusammen. In Version 3.6 gehören dazu die Typen 5, 6, 16 und 17.

Physical Memory Array

DMI-Typ 16 beschreibt ein physisches Speicherarray.

Dort können beispielsweise Angaben zur möglichen Gesamtkapazität oder zur Anzahl von Speichergeräten enthalten sein.

Memory Device

DMI-Typ 17 beschreibt einzelne Speichergeräte beziehungsweise Speichersteckplätze.

Je nach Firmware können dort Informationen erscheinen wie:

Größe,

Formfaktor,

Locator,

Typ,

Geschwindigkeit,

Hersteller,

Seriennummer,

Part Number

und weitere Speichereigenschaften.

Leere Speicherbänke

SMBIOS kann auch Steckplätze beziehungsweise Memory Devices beschreiben, in denen aktuell kein Speichermodul installiert ist.

Damit kann dmidecode sowohl:

vorhandene Module

als auch:

von der Firmware beschriebene Speicherpositionen

anzeigen.

Wichtig: maximale Kapazität ist nicht installierter Speicher

Die DMI-Tabelle kann nicht nur die aktuelle Konfiguration beschreiben, sondern auch mögliche Ausbauzustände.

Die dmidecode-Dokumentation nennt ausdrücklich als Beispiele:

die schnellste unterstützte CPU

oder:

die maximal unterstützte Speichermenge.

Deshalb muss zwischen:

installiert

und:

laut Firmware unterstützt

unterschieden werden.

RAM-Daten sind kein Speichertest

Wenn dmidecode ein Speichermodul meldet, bedeutet dies nicht, dass das Modul fehlerfrei arbeitet.

dmidecode führt keinen Speicherbelastungs- oder Speicherfehlertest durch.

Einzelne Werte mit `--string`

Mit:

```
-s KEYWORD
```

beziehungsweise:

```
--string KEYWORD
```

kann ein einzelner definierter DMI-Wert ausgegeben werden.

Systemhersteller

```
dmidecode -s system-manufacturer
```

Systemmodell

```
dmidecode -s system-product-name
```

Systemversion

```
dmidecode -s system-version
```

Seriennummer

```
dmidecode -s system-serial-number
```

System-UUID

```
dmidecode -s system-uuid
```

BIOS-Hersteller

```
dmidecode -s bios-vendor
```

BIOS-Version

```
dmidecode -s bios-version
```

BIOS-Datum

```
dmidecode -s bios-release-date
```

Mainboard-Hersteller

```
dmidecode -s baseboard-manufacturer
```

Mainboard-Modell

```
dmidecode -s baseboard-product-name
```

Mainboard-Seriennummer

```
dmidecode -s baseboard-serial-number
```

Gehäuseinformationen

Version 3.6 unterstützt unter anderem:

`chassis-manufacturer`

`chassis-type`

`chassis-version`

`chassis-serial-number`

und:

`chassis-asset-tag`.

Prozessorinformationen

Zu den unterstützten String-Schlüsseln gehören:

`processor-family`

`processor-manufacturer`

`processor-version`

und:

`processor-frequency`.

Mehrere Prozessoren

Ein einzelnes String-Schlüsselwort kann auf Mehrprozessorsystemen mehrere Ergebnisse liefern.

Die Dokumentation nennt `processor-version` ausdrücklich als Beispiel.

Verfügbare String-Schlüssel anzeigen

Version 3.6 führte:

```
--list-strings
```

ein.

Damit können die für:

```
--string
```

verfügbaren Schlüsselwörter aufgelistet werden.

Verfügbare Typen anzeigen

Ebenfalls mit Version 3.6 eingeführt wurde:

```
--list-types
```

Damit können die verfügbaren Schlüsselwörter für:

```
--type
```

aufgelistet werden.

Bestimmten Handle anzeigen

Mit:

```
-H HANDLE
```

beziehungsweise:

```
--handle HANDLE
```

kann genau der DMI-Eintrag angezeigt werden, dessen 16-Bit-Handle der angegebenen Kennung entspricht.

Nutzen der Handles

Handles sind insbesondere für detaillierte technische Analysen nützlich, wenn ein DMI-Eintrag auf einen anderen verweist.

Damit können Beziehungen innerhalb der SMBIOS-Struktur nachvollzogen werden.

Quiet-Modus

Mit:

```
-q
```

beziehungsweise:

```
--quiet
```

wird die Ausgabe reduziert.

Dabei werden laut Dokumentation unter anderem:

unbekannte,

inaktive

und OEM-spezifische Einträge

nicht angezeigt.

Außerdem werden Metadaten und Handle-Referenzen verborgen.

Rescue-Einsatz von `--quiet`

Für einen schnellen lesbaren Bericht kann der Quiet-Modus hilfreich sein.

Für eine vollständige technische Untersuchung sollte jedoch berücksichtigt werden, dass dadurch Informationen bewusst ausgeblendet werden.

Rohdaten als Hexdump

Mit:

`-u`

beziehungsweise:

`--dump`

werden die Einträge nicht normal decodiert, sondern hexadezimal ausgegeben.

Angehängte Strings erscheinen sowohl hexadezimal als auch als ASCII.

Die Funktion ist laut Dokumentation primär für Debugging vorgesehen.

Binäre DMI-Sicherung

Mit:

`--dump-bin FILE`

kann die DMI-Struktur binär in eine Datei geschrieben werden.

Die erzeugte Datei kann später wieder mit:

`--from-dump FILE`

eingelesen werden.

Wichtige Dateiregel

Für:

`--dump-bin FILE`

verlangt dmidecode 3.6, dass die Zieldatei noch nicht existiert.

Offline-Auswertung

Ein sinnvoller Ablauf kann damit lauten:

```
dmidecode --dump-bin system-dmi.bin
```

Anschließend kann die gespeicherte DMI-Struktur später mit:

```
dmidecode --from-dump system-dmi.bin
```

erneut decodiert werden.

Nutzen für Rescue-Dokumentation

Damit kann der zum Zeitpunkt der Untersuchung gelesene SMBIOS-/DMI-Datenbestand gesichert und später ausgewertet werden.

Das ist besonders hilfreich, wenn:

der Rechner später nicht mehr verfügbar ist,

Hardware ausgetauscht wird

oder die DMI-Daten separat archiviert werden sollen.

DMI-Dump ist kein Festplattenabbild

Eine mit `--dump-bin` erzeugte Datei enthält die DMI-Daten.

Sie enthält nicht:

Festplatteninhalte,

Arbeitsspeicherinhalte,

Dateisysteme

oder ein vollständiges Firmware-Abbild.

OEM Strings

Mit:

```
--oem-string N
```

kann ein bestimmter OEM-String ausgegeben werden.

Die Nummerierung beginnt bei 1.

Mit:

```
--oem-string count
```

kann die Anzahl der vorhandenen OEM-Strings ermittelt werden.

OEM-Daten

OEM-spezifische Einträge können herstellerabhängige Informationen enthalten.

dmidecode kann solche Daten nur vollständig decodieren, wenn entsprechende Herstellerinformationen beziehungsweise Unterstützung vorhanden sind.

`--no-quirks`

Mit:

`--no-quirks`

werden die Tabellenwerte möglichst direkt decodiert, ohne bekannte Firmwarefehler zu korrigieren oder irrelevante Felder auszublenden.

Die Dokumentation beschreibt diesen Modus primär als Funktion für Firmwareentwickler.

Rescue-Einordnung von `--no-quirks`

Für eine normale Hardwareinventarisierung ist:

`--no-quirks`

nicht erforderlich.

Die Option ist vielmehr hilfreich, wenn die unveränderte Firmwaredarstellung selbst Gegenstand der Untersuchung ist.

`--no-sysfs`

Mit:

`--no-sysfs`

wird dmidecode angewiesen, die DMI-Daten nicht über sysfs zu lesen.

Diese Option ist laut Dokumentation hauptsächlich für Debugging vorgesehen.

Firmwarequalität ist entscheidend

Die Zuverlässigkeit der Ausgabe hängt wesentlich davon ab, welche Daten der Hersteller in SMBIOS eingetragen hat.

Mögliche problematische Werte sind beispielsweise:

fehlende Seriennummern,

generische Modellbezeichnungen,

nicht gepflegte Asset-Tags,

falsche RAM-Angaben,

Platzhalter

oder unvollständige Tabellen.

Typische Platzhalter

Auf realen Systemen können Hersteller statt sinnvoller Daten generische Werte hinterlegen.

Solche Werte dürfen nicht allein aufgrund ihrer Ausgabe durch dmidecode als verifizierte Hardwareidentität behandelt werden.

Firmwareangabe und physische Beschriftung können abweichen

Bei kritischen Identifikationsaufgaben sollten wichtige Daten gegebenenfalls mit weiteren Quellen abgeglichen werden.

Beispielsweise:

Gehäuseetikett,

Mainboardbeschriftung,

Datenträgeretikett,

HWInfo,

LSHW

oder herstellerspezifische Diagnoseinformationen.

Seriennummern

Seriennummern sind für Inventarisierung und Support besonders wertvoll.

Sie sind gleichzeitig potenziell sensible Identifikatoren.

Ein vollständiger dmidecode-Bericht sollte deshalb nicht unkontrolliert veröffentlicht werden.

System-UUID

Dasselbe gilt für die:

System UUID.

Sie kann ein System eindeutig beziehungsweise langfristig identifizierbar machen.

Asset-Tags

Asset-Tags können interne Unternehmenskennzeichnungen enthalten.

Auch sie sollten vor externer Weitergabe eines Berichts geprüft werden.

Datenschutz

Anders als Inxi `-z` oder LSHW `-sanitize` besitzt dmidecode keinen allgemeinen Sanitizing-Schalter, der einen vollständigen Bericht automatisch für eine öffentliche Weitergabe bereinigt.

Deshalb gilt:

dmidecode-Ausgaben vor Weitergabe manuell prüfen und erforderlichenfalls gezielt reduzieren.

Besser gezielt abfragen

Wenn nur die BIOS-Version benötigt wird, ist:

```
dmidecode -s bios-version
```

datensparsamer als die Weitergabe eines vollständigen:

```
dmidecode
```

Reports.

BIOS-/Firmwarediagnose

Ein sinnvoller Firmwareüberblick kann beispielsweise mit:

```
dmidecode --type bios
```

erzeugt werden.

Damit können BIOS-bezogene DMI-Einträge gezielt untersucht werden.

BIOS-Version ist keine Firmwareintegritätsprüfung

dmidecode kann eine gemeldete Firmwareversion anzeigen.

Es überprüft damit jedoch nicht:

die kryptografische Integrität der Firmware,

deren Sicherheit,

ob Schadcode vorhanden ist

oder ob die Firmware tatsächlich dem unveränderten Herstellerstand entspricht.

Mainboarddiagnose

Für Mainboardinformationen:

```
dmidecode --type baseboard
```

Damit können die von SMBIOS gemeldeten Boardinformationen gezielt betrachtet werden.

Systemidentifikation

Ein schneller Identifikationsablauf kann lauten:

```
dmidecode -s system-manufacturer
dmidecode -s system-product-name
dmidecode -s system-serial-number
```

Datenschutz beachten

Die dritte Abfrage liefert eine Seriennummer, soweit sie vorhanden ist.

Diese sollte nicht ohne Not in öffentlich weitergegebenen Screenshots oder Supportbeiträgen erscheinen.

RAM-Inventarisierung

Für eine ausführliche Speicherübersicht:

```
dmidecode --type memory
```

Damit können die von der Firmware beschriebenen Speicherarrays und Speichermodule untersucht werden.

Typischer Rescue-Fall: RAM-Aufrüstung

dmidecode kann Hinweise liefern auf:

vorhandene Speicherbänke,

belegte beziehungsweise beschriebene Slots,

installierte Modulgrößen

und laut SMBIOS unterstützte Kapazitäten.

Vor einem tatsächlichen Hardwarekauf sollten diese Angaben jedoch mit der Herstellerdokumentation des konkreten Systems abgeglichen werden.

Grund

SMBIOS kann:

fehlerhaft,

unvollständig

oder veraltet

sein.

Typischer Rescue-Fall: unbekannter Rechner

1. Systemhersteller

```
dmidecode -s system-manufacturer
```

2. Modell

```
dmidecode -s system-product-name
```

3. BIOS

```
dmidecode --type bios
```

4. Mainboard

```
dmidecode --type baseboard
```

5. Speicher

```
dmidecode --type memory
```

6. Ergebnisse mit HWInfo, Inxi oder LSHW abgleichen

Typischer Rescue-Fall: BIOS-Update vorbereiten

Vor einem Firmwareupdate können beispielsweise:

Systemhersteller,

Systemmodell,

Mainboardmodell

und:

aktuelle BIOS-Version

ermittelt werden.

Das Firmwarepaket selbst muss anschließend ausschließlich anhand der eindeutigen Herstellerinformationen für das konkrete System ausgewählt werden.

Kritischer Sicherheitshinweis

Eine vermeintlich passende BIOS-Bezeichnung aus dmidecode allein rechtfertigt kein Firmwareupdate.

Vor einem schreibenden Firmwarevorgang müssen Modell, Hardwarevariante und Herstellerfreigabe eindeutig geprüft werden.

Typischer Rescue-Fall: Hardware vor Austausch dokumentieren

Vor dem Austausch von:

Mainboard,

RAM,

CPU

oder vollständigem System

kann ein DMI-Dump gespeichert werden:

```
dmidecode --dump-bin system-before.bin
```

Zusätzlich kann eine menschenlesbare Ausgabe archiviert werden.

Nach Hardwareänderung

Nach einem Austausch kann erneut ein Bericht erzeugt und mit dem vorherigen Zustand verglichen werden.

Dabei ist zu beachten, dass Firmware- und Herstellerdaten nicht zwangsläufig jede physische Änderung korrekt widerspiegeln.

Typischer Rescue-Fall: Systembericht

Für einen internen Bericht kann die vollständige Ausgabe in eine Datei geschrieben werden:

```
dmidecode > dmidecode-report.txt
```

Vor einer externen Weitergabe müssen sensible Werte entfernt beziehungsweise der Bericht auf die tatsächlich benötigten Informationen reduziert werden.

Typischer Rescue-Fall: nur BIOS-Version benötigt

Statt des vollständigen Reports:

```
dmidecode -s bios-version
```

Das ist übersichtlicher und datensparsamer.

Typischer Rescue-Fall: nur Systemmodell benötigt

```
dmidecode -s system-product-name
```

Typischer Rescue-Fall: Speicherproblem

Ein sinnvoller Diagnosepfad lautet:

1.

```
dmidecode --type memory
```

Firmwarebeschreibung der Speicherbestückung erfassen.

2.

mit LSHW und Inxi gegenprüfen.

3.

tatsächlich vom Kernel verfügbaren Arbeitsspeicher betrachten.

4.

bei Verdacht auf RAM-Fehler einen echten Speichertest durchführen.

dmidecode und HWInfo

HWInfo

erkennt Hardware über verschiedene System- und Hardwarequellen.

dmidecode

decodiert gezielt die von der Firmware bereitgestellte SMBIOS-/DMI-Tabelle.

dmidecode und Inxi

Inxi

integriert viele Systeminformationen in einen lesbaren Gesamtbericht.

dmidecode

liefert die wesentlich detailliertere direkte Sicht auf SMBIOS/DMI.

dmidecode und LSHW

LSHW

baut einen hierarchischen Hardwarebaum aus mehreren Informationsquellen auf.

dmidecode

konzentriert sich auf die Firmwaretabellen.

LSHW kann selbst DMI als eine seiner Informationsquellen verwenden.

Widersprüche zwischen Werkzeugen

Wenn beispielsweise:

dmidecode,

LSHW

und:

HWInfo

unterschiedliche Angaben liefern, sollte nicht automatisch die DMI-Ausgabe als maßgeblich betrachtet werden.

Stattdessen muss untersucht werden:

Welche Quelle liefert die jeweilige Information?

Physische Realität vor Firmwareangabe

Für kritische Rescue-Entscheidungen gilt:

Tatsächlich erkannte beziehungsweise physisch verifizierte Hardware hat Vorrang vor einer unbestätigten Firmwarebeschreibung.

SMBIOS-Unterstützung von Version 3.6

dmidecode 3.6 wurde am 24. April 2024 veröffentlicht.

Diese Version ergänzte Unterstützung für:

SMBIOS 3.6.0

und:

SMBIOS 3.7.0.

Dazu kamen unter anderem neue Speichergerätetypen, Prozessor-Upgrades, Porttypen, Slot-Eigenschaften und weitere Speicherfelder.

Neue Optionen in 3.6

Version 3.6 führte außerdem:

`--list-strings`

und:

`--list-types`

ein.

Neuere SMBIOS-Version als dmidecode unterstützt

Wenn die Firmware eine SMBIOS-Version meldet, die von der eingesetzten dmidecode-Version noch nicht unterstützt wird, gibt dmidecode einen entsprechenden Hinweis aus.

Die Dokumentation warnt in diesem Fall davor, die ausgegebenen Daten als zuverlässig anzusehen.

Rescue-Konsequenz

Eine solche Warnung darf nicht ignoriert werden.

Der Bericht sollte dann entsprechend gekennzeichnet und mit anderen Hardwarequellen abgeglichen werden.

Grenzen

dmidecode liest SMBIOS-/DMI-Daten.

dmidecode führt keinen Hardware-Stresstest durch.

dmidecode prüft RAM nicht auf Speicherfehler.

dmidecode prüft CPU-Stabilität nicht.

dmidecode prüft Datenträgergesundheit nicht.

dmidecode ersetzt SMART nicht.

dmidecode ersetzt NVMe-Diagnose nicht.

dmidecode misst keine Temperaturen.

dmidecode verifiziert Herstellerangaben nicht physisch.

dmidecode beweist nicht, dass ein von SMBIOS beschriebener Erweiterungszustand tatsächlich praktisch nutzbar ist.

dmidecode prüft keine Firmwareintegrität.

Firmwaredaten können falsch oder unvollständig sein.

Sicherheitshinweise

DMI-Daten nicht ungeprüft als physisch verifizierte Hardwaredaten behandeln.

Seriennummern, UUIDs und Asset-Tags als sensible Identifikatoren behandeln.

Vollständige Berichte nicht unkontrolliert veröffentlichen.

Für externe Supportfälle möglichst nur die tatsächlich benötigten Informationen ausgeben.

Firmwareupdates niemals allein aufgrund einer einzelnen dmidecode-Angabe auswählen.

Speicherinformationen nicht mit einem RAM-Funktionstest verwechseln.

Bei widersprüchlichen Angaben weitere Hardwarewerkzeuge zur Gegenprüfung verwenden.

Warnungen über nicht unterstützte SMBIOS-Versionen ernst nehmen.

Technische Einordnung in JUMBO BLOCK Rescue

dmidecode ist das vierte Programm des Hardware-Blocks:

A.29 HWInfo

breite Hardwareerkennung.

↓

A.30 Inxi

konsolidierter Systembericht.

↓

A.31 LSHW

hierarchische Hardware- und Busstruktur.

↓

A.32 DMI Decode

Firmwarebeschreibung über SMBIOS/DMI.

↓

A.33 Sensoren

aktuelle Messwerte unterstützter Hardware-Sensoren.

Die vier bisher behandelten Werkzeuge betrachten dasselbe System aus unterschiedlichen Perspektiven:

HWInfo

Was erkennt die Rescue-Umgebung?

Inxi

Wie sieht die laufende Systemkonfiguration zusammengefasst aus?

LSHW

Wie sind Geräte und Busse strukturell zugeordnet?

dmidecode

Was meldet die Firmware über System und Hardware?

Gerade die Kombination dieser Perspektiven ist wertvoll.

Stimmen:

Hersteller,

Systemmodell,

Mainboard,

CPU

und Speicherinformationen

über mehrere unabhängige Informationswege überein, entsteht eine wesentlich belastbarere Inventarisierung als durch eine einzelne Quelle.

Die zentrale Sicherheitsregel von DMI Decode lautet deshalb:

SMBIOS/DMI ist eine wertvolle Informationsquelle – aber keine unabhängige physische Verifikation der Hardware.

A.33 Sensoren / lm-sensors

Bereich: Hardware
Programm: lm-sensors
Baseline 20261001: lm-sensors 3.6.2
Installiertes Paket: `lm-sensors 1:3.6.2-2`
Zugehörige Bibliothek: `libsensors5 1:3.6.2-2`
Konfigurationspaket: `libsensors-config 1:3.6.2-2`
Bedienung: Kommandozeile
Aufgabe: Vom Linux-Hardware-Monitoring-System bereitgestellte Sensorwerte erkennen und anzeigen

Zweck

lm-sensors ist eine Werkzeugsammlung für Hardware-Monitoring unter Linux.

Das zentrale Kommando:

`sensors`

zeigt die aktuellen Messwerte der erkannten und vom System unterstützten Sensorchips an.

Abhängig von Hardware, Kernel-Treibern und Sensorunterstützung können dazu beispielsweise gehören:

CPU-Temperaturen,

Mainboard-Temperaturen,

weitere Temperatursensoren,

Lüfterdrehzahlen,

Spannungen,

Stromwerte,

Leistungswerte

und weitere von Hardware-Monitoring-Treibern bereitgestellte Messgrößen.

Welche Werte tatsächlich verfügbar sind, hängt vollständig vom jeweiligen System ab.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue dient lm-sensors insbesondere dazu:

aktuelle Temperaturwerte zu kontrollieren,

auffällige thermische Zustände zu erkennen,

verfügbare Lüfterdrehzahlen zu prüfen,

vom System bereitgestellte Spannungswerte einzusehen,

Hardwareprobleme mit möglichem Temperaturbezug zu untersuchen,

die thermische Situation vor längeren Rescue-Vorgängen einzuschätzen

und Sensorinformationen mit anderen Hardwarediagnosen zu kombinieren.

Zentrale Abgrenzung

lm-sensors ist ein:

Hardware-Monitoring-Werkzeug.

Es ist kein vollständiger Hardwaretest.

Ein Sensorwert beantwortet zunächst nur:

Welchen Messwert stellt der betreffende Sensor beziehungsweise Treiber aktuell bereit?

Er beantwortet nicht automatisch:

Ist die gesamte Hardware fehlerfrei?

Drei getrennte Ebenen

Für die Arbeit mit lm-sensors sollten drei Ebenen unterschieden werden:

1. Sensorhardware

Ein physischer Sensor beziehungsweise Monitoring-Chip muss vorhanden sein.

2. Kernel-Unterstützung

Ein geeigneter Treiber muss den Sensor erkennen und seine Daten bereitstellen.

3. lm-sensors

`sensors` liest die bereitgestellten Werte aus und stellt sie lesbar dar.

Fehlt eine dieser Ebenen, kann ein erwarteter Messwert fehlen.

Grundaufruf

Der wichtigste Aufruf lautet:

```
sensors
```

Er zeigt die aktuellen Messwerte aller von libsensors gefundenen Sensorchips an.

Typische Ausgabe

Abhängig vom System kann eine Ausgabe beispielsweise Kategorien enthalten wie:

Temperatur,

Lüfterdrehzahl,

Spannung,

Leistung

oder andere hardwareabhängige Werte.

Die konkrete Ausgabe unterscheidet sich erheblich zwischen verschiedenen Rechnern.

Keine feste Sensorliste erwarten

Es gibt keine universelle Ausgabe, die auf jedem Computer identisch vorhanden sein muss.

Ein Rechner kann beispielsweise:

nur CPU-Temperaturen,

CPU- und Mainboardwerte,

zahlreiche Spannungs- und Lüfterwerte

oder überhaupt keine durch lm-sensors nutzbaren Werte

bereitstellen.

Temperaturen

Temperaturen werden standardmäßig in Grad Celsius ausgegeben.

Mit:

```
sensors -f
```

beziehungsweise:

```
sensors --fahrenheit
```

werden Temperaturwerte stattdessen in Fahrenheit dargestellt.

Celsius für technische Dokumentation

Für JUMBO BLOCK Rescue sollte im Regelfall die standardmäßige Celsius-Ausgabe verwendet werden.

Dies erleichtert:

Vergleich,

Dokumentation

und technische Kommunikation.

Chipauswahl

`sensors` kann auf bestimmte Sensorchips beziehungsweise Chipmuster beschränkt werden.

Das Grundschema lautet:

```
sensors [OPTIONEN] [CHIPS]
```

Damit muss bei komplexen Systemen nicht zwingend die gesamte Sensorausgabe betrachtet werden.

Adapterinformationen

Standardmäßig können zu Sensorchips auch Adapterinformationen erscheinen.

Mit:

```
sensors -A
```

beziehungsweise:

```
sensors --no-adapter
```

werden diese Adapterangaben ausgeblendet.

Nutzen

Dies kann eine kompaktere Darstellung erzeugen, wenn nur die Messwerte selbst relevant sind.

Rohdaten

Mit:

```
sensors -u
```

wird eine Rohdarstellung ausgegeben.

Dieser Modus ist laut Dokumentation insbesondere geeignet für:

Debugging,

Weiterverarbeitung durch Skripte

und:

Erstellung beziehungsweise Untersuchung einer Sensorkonfiguration.

Unterschied zur normalen Ausgabe

Die normale Ausgabe ist für Menschen aufbereitet.

`-u` zeigt dagegen die zugrunde liegenden Rohbezeichnungen und Werte.

Für einen normalen Rescue-Schnellcheck ist daher:

```
sensors
```

meist besser lesbar.

Für eine technische Detailanalyse kann:

```
sensors -u
```

hilfreich sein.

JSON-Ausgabe

lm-sensors unterstützt eine JSON-Ausgabe.

Damit können Sensordaten maschinell weiterverarbeitet werden.

Die Baseline-Dokumentation kennt hierfür entsprechende JSON-Ausgabeoptionen.

Nutzen strukturierter Ausgabe

Dies kann beispielsweise hilfreich sein für:

Skripte,

automatische Dokumentation,

Monitoring

oder spätere Datenverarbeitung.

Für eine manuelle Rescue-Diagnose genügt normalerweise die Standardausgabe.

Version

Mit:

```
sensors -v
```

beziehungsweise:

```
sensors --version
```

wird die Programmversion angezeigt.

Hilfe

Mit:

```
sensors -h
```

beziehungsweise:

```
sensors --help
```

werden die verfügbaren Optionen angezeigt.

Konfigurationsdateien

libsensors verwendet systemweite Konfigurationsdateien.

Dazu gehören:

`/etc/sensors3.conf`

und als ältere Alternative:

`/etc/sensors.conf`.

Zusätzliche Konfigurationsdateien können unter:

`/etc/sensors.d`

liegen.

Bedeutung der Konfiguration

Die Konfiguration kann unter anderem beeinflussen:

Bezeichnungen,

Berechnungen,

ignorierte Werte,

Grenzwerte

und weitere Darstellungs- beziehungsweise Sensoreigenschaften.

Alternative Konfigurationsdatei

Mit:

```
sensors -c DATEI
```

beziehungsweise:

```
sensors --config-file DATEI
```

kann eine bestimmte Konfigurationsdatei verwendet werden.

Konfiguration temporär umgehen

Die Dokumentation nennt:

```
sensors -c /dev/null
```

als Möglichkeit, die normale Standardkonfiguration vorübergehend nicht zu verwenden.

Dies kann bei der Fehlersuche hilfreich sein.

Rescue-Bedeutung

Wenn ein Sensorwert aufgrund einer Konfiguration:

umbenannt,

umgerechnet,

ignoriert

oder anderweitig verändert

erscheint, kann ein Vergleich mit der Rohdarstellung beziehungsweise ohne Standardkonfiguration bei der Diagnose helfen.

Sensorgrenzen

Sensorchips können je nach Hardware verschiedene Grenzwerte bereitstellen.

Dazu können beispielsweise gehören:

maximale Temperaturen,

kritische Temperaturen

oder Hysteresegrenzen.

Grenzwert ist nicht automatisch Herstellerfreigabe

Ein von `sensors` dargestellter Grenzwert darf nicht ungeprüft als allgemeine sichere Betriebsgrenze der gesamten Hardware interpretiert werden.

Der Wert kann:

vom Sensorchip,

vom Treiber,

von der Firmware

oder:

von der libsensors-Konfiguration

stammen.

Plausibilitätsprüfung

Ein angezeigter Messwert sollte deshalb immer im technischen Kontext bewertet werden.

Beispielsweise wäre eine offensichtlich unrealistische Spannung oder Temperatur zunächst ein Grund, die:

Sensorzuordnung,

Treiberunterstützung

und:

Konfiguration

zu überprüfen.

Falsche oder unverbundene Sensoreingänge

Hardware-Monitoring-Chips können Eingänge besitzen, die auf einem konkreten Mainboard:

nicht angeschlossen,

anders beschaltet

oder vom Hersteller anders verwendet

werden.

Ein ausgegebener Wert ist daher nicht allein deshalb physikalisch sinnvoll, weil `sensors` ihn anzeigen kann.

Lüfterdrehzahlen

Wenn entsprechende Sensoren und Treiber vorhanden sind, können Lüfterdrehzahlen angezeigt werden.

Typischerweise werden diese in:

RPM

angegeben.

Nullwert beim Lüfter

Eine angezeigte Drehzahl von null kann verschiedene Ursachen haben.

Beispielsweise:

Lüfter steht tatsächlich,

Lüfter wird temperaturabhängig abgeschaltet,

Sensorleitung fehlt,

falscher Eingang,

Treiber-/Konfigurationsproblem.

Deshalb darf ein einzelner Nullwert nicht ohne weitere Prüfung automatisch als Lüfterdefekt gewertet werden.

Spannungen

Unterstützte Hardware-Monitoring-Chips können Spannungswerte bereitstellen.

Auch hier gilt:

Plausibilität prüfen.

Ein unplausibler Wert kann aus einer falschen Skalierung oder nicht passenden Sensorkonfiguration resultieren.

Leistungs- und Stromwerte

Moderne Kernel-Hardware-Monitoring-Treiber können abhängig von der Plattform zusätzlich:

Leistung,

Strom

oder andere Messgrößen

bereitstellen.

Ob diese in `sensors` erscheinen, hängt von der jeweiligen Hardware- und Treiberunterstützung ab.

CPU-Sensoren

Viele moderne Prozessoren besitzen interne Temperatursensoren.

Sind die entsprechenden Kernel-Treiber aktiv, können diese Werte über lm-sensors erscheinen.

Mehrere CPU-Kerne

Abhängig von Prozessor und Treiber können:

Gesamtwerte,

Package-Werte

oder:

einzelne Kernwerte

angezeigt werden.

Nicht jede CPU stellt dieselbe Struktur bereit.

Sensorwerte sind Momentaufnahmen

Der Befehl:

```
sensors
```

zeigt aktuelle Werte.

Er ist kein historisches Monitoringarchiv.

Ein einmaliger unauffälliger Messwert beweist deshalb nicht, dass das System unter Last oder zu einem früheren Zeitpunkt ebenfalls unauffällig war.

Wiederholte Beobachtung

Für die Diagnose eines thermischen Problems kann es sinnvoll sein, die Werte über einen Zeitraum wiederholt zu betrachten.

Dabei muss jedoch beachtet werden:

lm-sensors selbst erzeugt durch das bloße Auslesen keinen Belastungstest.

Keine künstliche Last erforderlich

Für eine Rescue-Erstdiagnose sollte ein möglicherweise instabiles System nicht unnötig belastet werden, nur um Temperaturwerte steigen zu sehen.

Insbesondere bei:

ungeklärten Abstürzen,

Lüfterproblemen,

offensichtlicher Überhitzung

oder instabiler Stromversorgung

hat die sichere Datenrettung Vorrang vor einem Belastungsexperiment.

`sensors-detect`

Zum lm-sensors-Paket gehört:

`sensors-detect`.

Dieses Programm sucht nach Hardware-Monitoring-Chips, die von libsensors beziehungsweise der lm-sensors-Werkzeugsammlung unterstützt werden.

Was `sensors-detect` untersucht

Die Dokumentation nennt in dieser Reihenfolge:

Sensoren in CPUs, Southbridges und Memory-Controllern,

Sensoren in Super-I/O-Chips,

Monitoring-Chips über ISA-I/O-Ports

und:

Monitoring-Chips über SMBus beziehungsweise I²C.

Interaktiver Betrieb

```
sensors-detect
```

führt den Benutzer interaktiv durch die Erkennungsschritte.

Automatischer Betrieb

Mit:

```
sensors-detect --auto
```

werden automatisch die Standardantworten auf die Fragen verwendet.

Wichtiger Sicherheitshinweis zu `--auto`

`--auto` bedeutet nicht „garantiert sicher“.

Die Programmdokumentation warnt ausdrücklich, dass die interne Logik auch im automatischen Modus zu potenziell gefährlichen Hardware-Probes führen kann.

Grundsätzliches Risiko von `sensors-detect`

`sensors-detect` muss für bestimmte Erkennungen direkt auf Hardware zugreifen.

Da das Programm vor der Erkennung noch nicht sicher wissen kann, welcher Chip tatsächlich vorhanden ist, können Probezugriffe erfolgen.

Die Entwickler weisen ausdrücklich darauf hin, dass dies Probleme verursachen kann.

Genannt werden mögliche Folgen von:

SMBus-Lockups

bis hin zu:

in seltenen Fällen permanenten Hardwareschäden.

Herstellerwarnung

Die Dokumentation empfiehlt deshalb ausdrücklich:

`sensors-detect` nicht auf Produktionsservern auszuführen

und:

keine Erkennungsschritte zu erzwingen, die standardmäßig übersprungen würden, sofern man nicht genau weiß, was man tut.

Konsequenz für JUMBO BLOCK Rescue

`sensors-detect` ist **kein automatischer Standard-Erstschritt**.

Zuerst sollte immer geprüft werden:

`sensors`

Sind bereits sinnvolle Werte vorhanden, besteht häufig kein Anlass, zusätzliche Hardware-Probes durchzuführen.

Empfohlener Rescue-Ablauf

1. Zuerst nur lesen

`sensors`

2. Vorhandene Werte beurteilen

Werden CPU-, Board- oder andere plausible Sensoren bereits angezeigt?

3. Nur bei tatsächlich fehlender Sensorunterstützung überlegen, ob `sensors-detect` erforderlich ist

4. Risiko und Bedeutung des Zielsystems berücksichtigen

5. Keine erzwungenen Probe-Schritte ohne konkreten technischen Grund

Produktionssysteme

Gerade bei einem kritischen Server, einer KRITIS-Umgebung oder anderer wichtiger Hardware ist der Hinweis der Entwickler besonders relevant.

Ein Rescue-Werkzeug darf nicht allein deshalb ausgeführt werden, weil es verfügbar ist.

Rescue-Grundsatz

Diagnosezugriff muss zum Risiko des Systems passen.

Kernelmodule

Wenn `sensors-detect` einen unterstützten Monitoring-Chip erkennt, kann ein entsprechender Kernel-Treiber erforderlich sein, damit dessen Werte über das Linux-Hardware-Monitoring-System verfügbar werden.

Live-System-Perspektive

Im JUMBO BLOCK Rescue Betrieb gelten die:

Kernelversion,

geladenen Treiber

und:

Hardwareunterstützung

der gestarteten Rescue-Umgebung.

Ein Sensor, der im installierten Zielbetriebssystem verfügbar war, muss deshalb nicht zwangsläufig auch in der Rescue-Umgebung identisch erscheinen – und umgekehrt.

Fehlender Sensorwert

Wenn ein erwarteter Wert fehlt, kann das beispielsweise bedeuten:

Sensor nicht vorhanden,

Sensor vom Kernel nicht unterstützt,

erforderlicher Treiber nicht geladen,

Hardware nicht zugänglich,

Sensor von der Konfiguration ignoriert

oder:

Plattform stellt den Wert auf anderem Weg bereit.

Fehlender Wert ist kein Beweis für fehlenden Sensor

Aus:

„lm-sensors zeigt diesen Wert nicht“

darf deshalb nicht automatisch geschlossen werden:

„Dieser Sensor existiert physisch nicht.“

`sensors -s`

Mit:

```
sensors -s
```

beziehungsweise:

```
sensors --set
```

werden die in der Konfigurationsdatei enthaltenen `set`-Anweisungen ausgewertet.

Hierfür sind Root-Rechte erforderlich.

Wichtige Abgrenzung

`sensors -s` ist kein reiner Lesevorgang.

Es kann entsprechend der Konfiguration Werte beziehungsweise Grenzparameter auf unterstützter Hardware setzen.

Rescue-Regel

Für eine reine Diagnose ist:

`sensors -s`

nicht erforderlich.

Es sollte nicht routinemäßig ausgeführt werden.

Konfigurationsanweisungen

`sensors.conf` unterstützt unter anderem:

`label`

`compute`

`ignore`

und:

`set`.

`label`

Damit können Sensorbezeichnungen angepasst werden.

`compute`

Damit können Rohwerte mathematisch umgerechnet werden.

`ignore`

Damit können bestimmte Sensorwerte ausgeblendet werden.

`set`

Damit können auf unterstützter Hardware Werte beziehungsweise Grenzparameter gesetzt werden.

Konsequenz

Die angezeigte Sensorbezeichnung oder ein berechneter Wert kann von der Konfiguration beeinflusst sein.

Das ist ein weiterer Grund, Sensorwerte nicht ohne Kontext zu interpretieren.

Temperatur-Sensortypen

Die Konfiguration unterstützt bei geeigneten Chips unterschiedliche Temperatursensortypen.

Die Dokumentation nennt unter anderem:

thermische Dioden,

Thermistoren,

AMD AMDSI

und:

Intel PECI.

Nicht jeder Chip unterstützt die Änderung dieser Sensortypen.

Firmwarekonfiguration

Die Dokumentation weist darauf hin, dass das BIOS den Sensortyp theoretisch korrekt konfigurieren sollte, dies in der Praxis jedoch nicht immer der Fall ist.

Hysterese

Temperaturgrenzen können zusätzlich Hysteresewerte besitzen.

Das bedeutet:

Ein Alarm kann bei einer oberen Temperatur ausgelöst werden und erst bei einer etwas niedrigeren Temperatur wieder zurückgesetzt werden.

Damit wird ein ständiges Ein-/Ausschalten einer Kühlreaktion unmittelbar am Grenzwert verhindert.

Sensoralarm

Je nach Hardware können Sensorwerte einen Alarmstatus anzeigen.

Ein solcher Alarm sollte ernst genommen und im Kontext des betreffenden Sensors untersucht werden.

Alarm bedeutet nicht automatisch Totalschaden

Ein Alarm kann auf:

überschrittenen Grenzwert,

fehlerhafte Konfiguration,

unplausiblen Sensoreingang

oder ein tatsächliches Hardwareproblem

hinweisen.

Die Ursache muss geklärt werden.

`--bus-list`

Mit:

```
sensors --bus-list
```

können Bus-Anweisungen erzeugt werden, die für die Sensorkonfiguration geeignet sind.

Dies ist insbesondere relevant, wenn mehrere Chips mit derselben Adresse auf unterschiedlichen Bussen vorhanden sind.

Busnummern

Die Dokumentation weist darauf hin, dass Busnummern über Neustarts hinweg nicht zwingend stabil bleiben.

Die Bus-Konfiguration ermöglicht deshalb die Referenzierung anhand von Busnamen statt ausschließlich anhand numerischer Busnummern.

Typischer Rescue-Fall: Rechner schaltet sich unerwartet ab

1.

```
sensors
```

ausführen.

2.

CPU- und gegebenenfalls Mainboardtemperaturen prüfen.

3.

Lüfterwerte kontrollieren, soweit verfügbar.

4.

auffällige Alarmmeldungen beachten.

5.

Werte mit Herstellerangaben und weiterer Hardwarediagnose abgleichen.

6.

bei offensichtlicher Überhitzung keine unnötige zusätzliche Last erzeugen.

Typischer Rescue-Fall: Lüfter läuft ständig auf hoher Drehzahl

```
sensors
```

kann helfen festzustellen:

ob hohe Temperaturen gemeldet werden

und:

welche Lüfterdrehzahlen verfügbar sind.

Danach muss zwischen möglichen Ursachen unterschieden werden, beispielsweise:

tatsächliche hohe Temperatur,

Kühlproblem,

Firmwaresteuerung,

Treiberproblem

oder falsche Sensorinterpretation.

Typischer Rescue-Fall: Lüfter steht

Ein angezeigter:

```
0 RPM
```

Wert allein genügt nicht für die Diagnose.

Zusätzlich prüfen:

Dreht der Lüfter physisch?

Ist Zero-RPM-Betrieb vorgesehen?

Ist der richtige Sensor zugeordnet?

Ist der Anschluss überhaupt tachometrisch überwacht?

Typischer Rescue-Fall: ungewöhnlich hohe CPU-Temperatur

1. Wert erneut prüfen

2. Package-/Core-Zuordnung beachten

3. Alarm- beziehungsweise Grenzwertinformationen prüfen

4. physische Kühlung kontrollieren

Lüfter,

Kühlkörper,

Luftstrom.

5. keine unnötige Belastung erzeugen

6. wichtige Daten sichern, wenn Systemstabilität fraglich ist

Typischer Rescue-Fall: offensichtlich falsche Spannung

Ein Wert, der technisch nicht plausibel erscheint, sollte nicht sofort als defektes Netzteil interpretiert werden.

Zunächst prüfen:

Sensorlabel,

Rohwert mit `sensors -u`,

Konfiguration,

Treiber

und:

Mainboardunterstützung.

Für eine belastbare elektrische Messung ist gegebenenfalls geeignete externe Messtechnik erforderlich.

Typischer Rescue-Fall: keine Sensoren gefunden

Wenn `sensors` keine nutzbaren Sensorwerte findet:

1. nicht automatisch Hardwaredefekt annehmen

2. Hardwaremodell mit HWInfo, Inxi, LSHW und dmidecode bestimmen

3. Kernel-/Treiberunterstützung prüfen

4. erst dann entscheiden, ob `sensors-detect` erforderlich und vertretbar ist

Typischer Rescue-Fall: thermische Kontrolle vor ddrescue

Ein länger laufender ddrescue-Vorgang kann Rechner und Datenträger über längere Zeit in Betrieb halten.

Vor Beginn kann ein Sensorcheck sinnvoll sein:

```
sensors
```

Damit kann zumindest festgestellt werden, ob bereits vor dem Vorgang auffällige Systemtemperaturen vorliegen.

Datenträgertemperatur

lm-sensors ist nicht das primäre Werkzeug zur Zustandsdiagnose von Festplatten oder NVMe-Laufwerken.

Für Datenträger sind:

A.8 SMART Diagnose

und:

A.9 NVMe Diagnose

maßgeblich.

Diese Werkzeuge können je nach Gerät auch laufwerksspezifische Temperaturdaten bereitstellen.

Sensorwert und SMART-Temperatur

CPU-/Mainboard-Sensorik und Datenträger-SMART/NVMe-Telemetrie stammen aus unterschiedlichen Quellen.

Sie sollten nicht miteinander verwechselt werden.

lm-sensors und HWInfo

HWInfo

identifiziert Hardware.

lm-sensors

zeigt unterstützte aktuelle Monitoringwerte.

lm-sensors und Inxi

Inxi

kann Sensorinformationen in seinen Gesamtbericht aufnehmen.

lm-sensors

ist das spezialisierte Werkzeug für die direkte Sensorausgabe.

lm-sensors und LSHW

LSHW

zeigt die Hardwarestruktur.

lm-sensors

zeigt aktuelle Messwerte unterstützter Monitoring-Hardware.

lm-sensors und dmidecode

dmidecode

zeigt firmwareseitige SMBIOS-/DMI-Beschreibungen.

lm-sensors

zeigt laufende Messwerte, die über Kernel-Hardware-Monitoring verfügbar sind.

Damit sind die beiden Werkzeuge grundsätzlich unterschiedlich.

Vier unterschiedliche Hardwareperspektiven

Der Hardware-Block von JUMBO BLOCK Rescue stellt damit verschiedene Sichtweisen bereit:

HWInfo

Hardwareerkennung.

Inxi

Systemübersicht.

LSHW

Hardwarehierarchie.

dmidecode

Firmwarebeschreibung.

lm-sensors

aktuelle Sensorwerte.

Schreibende Hilfsprogramme

Zum lm-sensors-Paket gehören neben `sensors` und `sensors-detect` weitere Low-Level-Werkzeuge.

Dazu gehört beispielsweise:

`isaset`.

Dieses Programm kann direkt ISA-Register verändern.

Sicherheitswarnung zu `isaset`

Solche Low-Level-Schreibwerkzeuge gehören **nicht zum normalen Sensor-Rescue-Workflow**.

Direkte Registerzugriffe mit falschen Parametern können:

Systemabstürze,

Datenverlust

oder weitere Schäden

verursachen.

`isadump`

Auch `isadump` greift auf ISA-Register zu.

Obwohl es primär Register liest, warnt seine Dokumentation ausdrücklich davor, mit beliebigen Parametern im ISA-Adressraum zu arbeiten.

JUMBO BLOCK Rescue Grundsatz

Low-Level-Sensorwerkzeuge werden nicht experimentell eingesetzt.

Für die normale Diagnose gilt:

`sensors` zuerst.

Keine automatischen Schreibvorgänge

Die reine Sensordiagnose benötigt normalerweise weder:

`sensors -s`

noch:

`isaset`

noch andere direkte Registeränderungen.

Grenzen

lm-sensors kann nur Sensoren anzeigen, die:

vorhanden,

zugänglich

und:

vom laufenden Linux-System unterstützt

werden.

Nicht jeder Computer stellt dieselben Sensoren bereit.

Nicht jeder Sensorwert ist automatisch korrekt beschriftet.

Nicht jeder angezeigte Grenzwert ist automatisch eine allgemeingültige Herstellergrenze.

Ein einzelner Sensorwert ist nur eine Momentaufnahme.

lm-sensors führt keinen CPU-Stresstest durch.

lm-sensors führt keinen RAM-Test durch.

lm-sensors ersetzt SMART nicht.

lm-sensors ersetzt NVMe-Diagnose nicht.

lm-sensors prüft keine Netzteile mit externer Messtechnik.

lm-sensors kann physische Kühlprobleme anzeigen helfen, repariert sie aber nicht.

`sensors-detect` ist nicht risikofrei.

Sicherheitshinweise

Für die Erstdiagnose zunächst ausschließlich:

`sensors`

verwenden.

`sensors-detect` nur einsetzen, wenn eine Sensorerkennung tatsächlich erforderlich ist.

Auf kritischen Produktionssystemen die ausdrückliche Warnung der Entwickler zu `sensors-detect` beachten.

Keine standardmäßig übersprungenen Detection-Schritte ohne konkreten technischen Grund erzwingen.

`--auto` nicht mit „risikofrei“ verwechseln.

`sensors -s` nicht als normalen Lesebefehl verwenden.

Low-Level-Schreibwerkzeuge wie `isaset` nicht experimentell einsetzen.

Unplausible Sensorwerte zuerst verifizieren, bevor daraus Hardwaredefekte abgeleitet werden.

Bei deutlicher Überhitzung keine unnötigen Belastungstests durchführen.

Bei instabilem System zuerst die Sicherung wichtiger Daten priorisieren.

Technische Einordnung in JUMBO BLOCK Rescue

Mit lm-sensors ist der Hardware-Block von JUMBO BLOCK Rescue vollständig:

A.29 HWInfo

Welche Hardware erkennt das System?

↓

A.30 Inxi

Wie sieht die laufende Systemkonfiguration insgesamt aus?

↓

A.31 LSHW

Wie ist die Hardware hierarchisch und über Busse strukturiert?

↓

A.32 DMI Decode

Was meldet BIOS/UEFI über System und Hardware?

↓

A.33 Sensoren / lm-sensors

Welche aktuellen Messwerte stellen unterstützte Hardware-Sensoren bereit?

Diese Werkzeuge ergänzen sich.

Ein typischer Diagnoseweg bei einem unbekannten oder instabilen Rechner kann lauten:

1. Inxi

schneller Gesamtüberblick.

2. HWInfo

gezielte Hardwareerkennung.

3. LSHW

Geräte- und Buszuordnung.

4. dmidecode

System-, Board-, BIOS- und Speicherinformationen aus SMBIOS.

5. sensors

aktuelle Temperatur- und Monitoringwerte.

Bei Storage-Problemen folgen:

SMART Diagnose

beziehungsweise:

NVMe Diagnose.

Bei einem physisch instabilen Datenträger anschließend gegebenenfalls:

GNU ddrescue.

Die zentrale Rolle von lm-sensors in JUMBO BLOCK Rescue lautet:

Den aktuellen thermischen und sensorischen Zustand des laufenden Systems sichtbar machen – ohne aus einem einzelnen Messwert mehr abzuleiten, als die Hardware und ihre Sensorik tatsächlich belegen.

Und für die Sensorerkennung gilt der Rescue-Grundsatz:

Erst vorhandene Werte lesen. Zusätzliche Hardware-Probes nur dann durchführen, wenn sie wirklich erforderlich und für das betreffende System vertretbar sind.

A.34 Midnight Commander

Bereich: Dateien
Programm: GNU Midnight Commander
Baseline 20261001: Midnight Commander 4.8.33
Installierte Pakete: `mc 3:4.8.33-1+deb13u1`, `mc-data 3:4.8.33-1+deb13u1`
Bedienung: textbasierte Zwei-Fenster-Oberfläche im Terminal
Aufgabe: Dateien und Verzeichnisse navigieren, anzeigen, kopieren, verschieben, umbenennen, suchen und bearbeiten sowie Shell-Kommandos aus einer gemeinsamen Oberfläche ausführen

Zweck

GNU Midnight Commander – kurz:

MC

– ist ein textbasierter Dateimanager beziehungsweise eine visuelle Shell für Unix-artige Systeme.

Die Oberfläche wird überwiegend von zwei Verzeichnisfenstern eingenommen. Dadurch können zwei Verzeichnisse gleichzeitig betrachtet und Dateioperationen zwischen ihnen durchgeführt werden.

Zusätzlich integriert Midnight Commander:

eine Shell-Kommandozeile,

einen internen Dateibetrachter

und einen internen Texteditor.

Einsatzgebiet im Rescue-System

Innerhalb von JUMBO BLOCK Rescue eignet sich Midnight Commander insbesondere zum:

Navigieren durch gemountete Dateisysteme,

Untersuchen geretteter Dateien,

Kopieren wichtiger Daten auf ein Sicherungsmedium,

Vergleichen von Quell- und Zielverzeichnissen,

Umbenennen und Organisieren geretteter Dateien,

Anzeigen von Text- und Konfigurationsdateien,

Bearbeiten von Konfigurationsdateien,

Suchen nach Dateien und Verzeichnissen,

Arbeiten mit unterstützten Archiven,

Aufrufen von Shell-Kommandos

und Verwalten von Dateien auf lokal oder anderweitig eingebundenen Dateisystemen.

Keine Datenrettungssoftware im engeren Sinn

Midnight Commander rekonstruiert keine verlorenen Partitionstabellen.

Er führt kein File Carving durch.

Er liest keine beschädigten Datenträger wie GNU ddrescue sektorweise aus.

Er stellt keine gelöschten Dateien aus Dateisystemmetadaten wieder her.

Seine Rolle beginnt typischerweise dort, wo ein Dateisystem bereits zugänglich ist oder Dateien bereits gerettet wurden.

Zwei-Panel-Prinzip

Der zentrale Arbeitsbereich besteht aus zwei Verzeichnisfenstern.

Ein Panel ist jeweils aktiv.

Das andere Panel kann beispielsweise das Ziel einer Dateioperation darstellen.

Dadurch lässt sich eine typische Rescue-Situation übersichtlich abbilden:

linkes Panel

Quellverzeichnis.

rechtes Panel

Sicherungsziel.

Oder umgekehrt.

Die Debian-Manpage bestätigt, dass Midnight Commander zwei Verzeichnisse gleichzeitig darstellt und Kopier- beziehungsweise Umbenennoperationen standardmäßig das Verzeichnis des anderen Panels als Ziel vorschlagen können. Vor der eigentlichen Operation erfolgt eine Bestätigung.

Programmstart

Midnight Commander wird gestartet mit:

```
mc
```

Start mit einem Verzeichnis

Ein bestimmtes Verzeichnis kann direkt angegeben werden:

```
mc /pfad/zum/verzeichnis
```

Dieses Verzeichnis erscheint dann im aktiven Panel.

Zwei Startverzeichnisse

Es können auch zwei Verzeichnisse angegeben werden:

```
mc /quelle /ziel
```

Das erste Verzeichnis wird im aktiven Panel angezeigt.

Das zweite Verzeichnis erscheint im anderen Panel.

Rescue-Nutzen

Damit kann beispielsweise unmittelbar eine Arbeitsansicht aufgebaut werden:

```
mc /mnt/quelle /mnt/sicherung
```

Voraussetzung ist selbstverständlich, dass diese Verzeichnisse tatsächlich existieren und die entsprechenden Dateisysteme korrekt eingebunden wurden.

Navigation

Mit den Pfeiltasten wird innerhalb eines Panels navigiert.

`Enter`

öffnet ein markiertes Verzeichnis.

Bei:

`..`

wechselt man in das übergeordnete Verzeichnis.

Zwischen den Panels wechseln

Mit:

`Tab`

wird zwischen den beiden Panels gewechselt.

Damit lässt sich schnell zwischen Quelle und Ziel wechseln.

Menü

Mit:

`F9`

wird die Menüleiste aktiviert.

Die Debian-Manpage nennt F9 ausdrücklich zum Anzeigen beziehungsweise Aktivieren der Menüleiste.

Funktionstasten

Am unteren Bildschirmrand zeigt Midnight Commander die aktuell wichtigen Funktionstasten an.

Die klassische Grundbelegung umfasst insbesondere:

`F1` – Hilfe

`F3` – Anzeigen

`F4` – Bearbeiten

`F5` – Kopieren

`F6` – Verschieben beziehungsweise Umbenennen

`F7` – Verzeichnis erstellen

`F8` – Löschen

`F9` – Menü

`F10` – Midnight Commander verlassen.

Rescue-Vorteil

Diese Darstellung macht viele Dateioperationen möglich, ohne sich sämtliche Shell-Kommandos merken zu müssen.

Interner Viewer

Mit:

`F3`

kann eine markierte Datei im internen Viewer geöffnet werden.

Der Viewer dient dem Betrachten von Dateien, ohne sie dafür mit einem Editor verändern zu müssen.

Rescue-Grundsatz

Wenn eine Datei zunächst nur untersucht werden soll:

Viewer vor Editor.

Das reduziert das Risiko unbeabsichtigter Änderungen.

Viewer direkt starten

Midnight Commander unterstützt außerdem:

```
mc -v DATEI
```

beziehungsweise:

```
mc --view=DATEI
```

zum direkten Start des internen Viewers mit einer bestimmten Datei.

Typische Rescue-Anwendungen des Viewers

Geeignet ist der Viewer beispielsweise für:

Logdateien,

Konfigurationsdateien,

Textdateien,

Skripte,

Listen,

Berichte

und andere lesbare Dateien.

Interner Editor

Mit:

`F4`

kann die markierte Datei im konfigurierten Editor geöffnet werden.

Midnight Commander enthält mit:

`mcedit`

einen eigenen internen Texteditor.

Editor direkt starten

Der interne Editor kann auch direkt gestartet werden:

```
mcedit DATEI
```

oder über Midnight Commander mit:

```
mc -e DATEI
```

beziehungsweise:

```
mc --edit=DATEI.
```

Bearbeiten bedeutet Schreiben

Sobald eine Datei gespeichert wird, wird ihr Inhalt verändert.

Für Rescue-Arbeiten gilt deshalb:

Dateien auf einem Beweissicherungs-, Original- oder beschädigten Quellmedium nicht ohne konkreten Grund bearbeiten.

Besser auf einer Arbeitskopie

Muss eine Konfigurationsdatei geändert werden, sollte vorher geklärt werden:

Ist das Originalmedium die richtige Arbeitsumgebung?

Existiert eine Sicherung?

Kann auf einer Kopie gearbeitet werden?

Ist die Änderung reversibel?

Kopieren

F5

öffnet die Kopierfunktion.

Das andere Panel dient dabei typischerweise als vorgeschlagenes Ziel.

Vor dem Kopiervorgang wird das Ziel bestätigt beziehungsweise kann angepasst werden.

Typischer Rescue-Fall

Linkes Panel:

`/mnt/beschaedigtes-system/home/benutzer`

Rechtes Panel:

`/mnt/backup/rettung`

Dann können ausgewählte Dateien beziehungsweise Verzeichnisse mit:

F5

auf das Sicherungsmedium kopiert werden.

Kopieren ist nicht automatisch Datenrettung

Wenn das Quelldateisystem oder der Datenträger physisch instabil ist, ist normales dateiweises Kopieren nicht zwangsläufig die beste erste Maßnahme.

Bei Lesefehlern oder sterbenden Datenträgern sollte zunächst geprüft werden, ob:

A.3 GNU ddrescue

die geeignetere Acquisition-Methode ist.

Warum?

Ein Dateimanager arbeitet auf Datei- und Verzeichnisebene.

GNU ddrescue arbeitet auf Blockebene und wurde speziell für problematische Lesemedien entwickelt.

Verschieben

F6

dient zum Verschieben beziehungsweise Umbenennen.

Wichtiger Rescue-Unterschied

Kopieren

erhält grundsätzlich die Quelle.

Verschieben

soll die Quelle nach erfolgreicher Operation nicht am ursprünglichen Ort belassen.

Für Rescue-Arbeiten ist deshalb bei wertvollen Originaldaten normalerweise:

Kopieren vor Verschieben

die sicherere Strategie.

Umbenennen

F6 kann ebenfalls zum Umbenennen von Dateien oder Verzeichnissen verwendet werden.

Auch dies verändert das Dateisystem.

Verzeichnis erstellen

Mit:

F7

kann ein neues Verzeichnis erstellt werden.

Dies ist beispielsweise auf einem Sicherungsmedium sinnvoll, um gerettete Daten strukturiert abzulegen.

Löschen

Mit:

F8

können ausgewählte Dateien beziehungsweise Verzeichnisse gelöscht werden.

Kritischer Sicherheitshinweis

Löschen ist im Rescue-Betrieb grundsätzlich eine Hochrisikooperation.

Vor dem Bestätigen muss eindeutig geklärt sein:

Welches Panel ist aktiv?

Welcher Datenträger ist dort eingehängt?

Welche Datei beziehungsweise welches Verzeichnis ist markiert?

Handelt es sich um Quelle oder Sicherungsziel?

Kein Papierkorb als Rescue-Strategie

Bei Terminal-Dateimanagern darf nicht davon ausgegangen werden, dass eine Löschoperation wie bei manchen Desktop-Dateimanagern sicher in einem Papierkorb landet.

Eine Löschung muss deshalb als potenziell endgültige Dateisystemänderung behandelt werden.

Mehrere Dateien auswählen

Midnight Commander unterstützt das Markieren mehrerer Dateien.

Damit können Dateioperationen auf Gruppen von Dateien angewendet werden.

Musterbasierte Auswahl

Dateien können auch anhand von Mustern ausgewählt beziehungsweise abgewählt werden.

Dies ist beispielsweise hilfreich, wenn nur bestimmte Dateitypen gesichert werden sollen.

Rescue-Vorsicht bei Massenauswahl

Vor:

Kopieren,

Verschieben

oder insbesondere:

Löschen

einer großen Auswahl sollte die markierte Dateimenge nochmals kontrolliert werden.

Dateisuche

Midnight Commander besitzt eine integrierte Suchfunktion.

Damit kann nach Dateien innerhalb eines Verzeichnisbaums gesucht werden.

Inhaltssuche

Die Suchfunktionen können außerdem bei der Suche nach Textinhalten in Dateien eingesetzt werden.

Das kann beispielsweise hilfreich sein bei der Suche nach:

Konfigurationswerten,

Hostnamen,

Fehlermeldungen,

Logeinträgen

oder bestimmten Zeichenketten.

Rescue-Nutzen

Auf einem zugänglichen, aber unbekannten Linux-System kann damit beispielsweise gezielt nach:

Konfigurationsdateien,

Logs,

Dokumenten

oder projektspezifischen Dateien

gesucht werden.

Suche ist kein File Carving

Midnight Commander sucht innerhalb der zugänglichen Dateisystemstruktur.

Gelöschte Dateien, die nicht mehr im Verzeichnisbaum vorhanden sind, werden dadurch nicht automatisch rekonstruiert.

Dafür sind je nach Situation:

TestDisk,

PhotoRec,

Foremost

oder:

Extundelete

zuständig.

Shell-Kommandozeile

Ein wichtiger Bestandteil von Midnight Commander ist die integrierte Shell-Kommandozeile.

Text, der dort eingegeben und mit:

`Enter`

bestätigt wird, wird als Shell-Kommando ausgeführt.

Bedeutung für Rescue

Damit können Dateimanager und Kommandozeile kombiniert werden.

Beispielsweise kann zunächst visuell zu einem Verzeichnis navigiert und anschließend dort ein Diagnosekommando ausgeführt werden.

Vorsicht

Die Shell-Kommandozeile besitzt die Rechte des gestarteten Midnight-Commander-Prozesses.

Ein dort eingegebener destruktiver Befehl bleibt destruktiv.

Midnight Commander macht Shell-Kommandos nicht automatisch sicherer.

Subshell

Midnight Commander unterstützt eine integrierte beziehungsweise parallele Subshell, sofern diese Funktion beim Build aktiviert wurde.

Version 4.8.33 brachte insbesondere Verbesserungen bei der Subshell-Unterstützung für bash, ash/dash und fish sowie erste Unterstützung für ksh-Varianten.

Rescue-Bedeutung

Damit kann zwischen Dateimanager und Shell-Arbeit gewechselt werden, ohne den gesamten Dateikontext zu verlieren.

Archive und virtuelles Dateisystem

Midnight Commander besitzt ein VFS-Konzept – ein virtuelles Dateisystem.

Dadurch können unterstützte Archive und andere Datenquellen teilweise so dargestellt werden, als wären sie Verzeichnisse.

Archive

Unterstützte Archivtypen können damit direkt aus der Dateimanageroberfläche untersucht werden, sofern die dafür erforderliche VFS-Unterstützung beziehungsweise externe Hilfsprogramme vorhanden sind.

Nicht mit echtem Mount verwechseln

Eine im Midnight Commander geöffnete Archivstruktur ist nicht automatisch ein regulär im Linux-Dateisystem gemountetes Dateisystem.

Andere Programme sehen diese interne VFS-Ansicht nicht zwangsläufig als normalen Mountpunkt.

Archiv nicht unnötig verändern

Bei Rescue- und Analysearbeiten sollten Archive zunächst nur betrachtet beziehungsweise auf ein Arbeitsmedium extrahiert werden.

Änderungen am Originalarchiv sind zu vermeiden, wenn dessen ursprünglicher Zustand erhalten werden soll.

Dateitypen und Aktionen

Midnight Commander kann abhängig von Dateityp und Konfiguration bestimmte Aktionen mit Dateien verknüpfen.

Wird auf einer Datei `Enter` gedrückt, kann daher abhängig von:

Dateityp,

Ausführbarkeit

und Konfiguration

eine Aktion oder ein externes Programm gestartet werden.

Rescue-Regel

Bei unbekannten oder potenziell schädlichen Dateien:

nicht unbedacht mit Enter ausführen.

Zum Untersuchen zunächst:

F3

verwenden.

Ausführbare Dateien

Ist die markierte Datei ausführbar, kann Enter sie ausführen.

Das ist insbesondere bei der Untersuchung eines möglicherweise kompromittierten Systems relevant.

Sicherheitsregel bei verdächtigen Dateien

Verdächtige Programme, Skripte oder Binärdateien nicht zur Analyse ausführen.

Das bloße Vorhandensein einer Datei ist kein Grund, sie zu starten.

Versteckte Dateien

Midnight Commander kann auch versteckte Unix-Dateien und -Verzeichnisse anzeigen.

Dies ist für Rescue-Arbeiten wichtig, da zahlreiche relevante Konfigurationen in Namen beginnen mit:

.

Beispiele sind Benutzerkonfigurationen innerhalb von Home-Verzeichnissen.

Dateirechte und Metadaten

Midnight Commander kann Unix-Dateiinformationen wie:

Berechtigungen,

Eigentümer,

Gruppe,

Größe

und Zeitangaben

darstellen beziehungsweise – abhängig von Operation und Rechten – bearbeiten.

Rescue-Bedeutung

Bei einer Wiederherstellung kann es wichtig sein, nicht nur Dateiinhalte, sondern auch:

Eigentümer,

Berechtigungen,

Zeitstempel

und symbolische Links

zu berücksichtigen.

Kopie ist nicht automatisch forensisch identisch

Eine normale Dateikopie mit einem Dateimanager ist nicht mit einem forensischen Blockabbild gleichzusetzen.

Abhängig von:

Dateisystem,

Kopiermethode,

Mountoptionen,

Berechtigungen

und Zielsystem

können Metadaten unterschiedlich behandelt werden.

Forensische Sicherung

Wenn eine unveränderte bitgenaue Sicherung des Quellmediums erforderlich ist, ist Midnight Commander nicht das geeignete Acquisition-Werkzeug.

Dafür wird ein Blockabbild beziehungsweise ein geeigneter forensischer Workflow benötigt.

Read-only-Quelle

Für Rescue-Arbeiten ist es häufig sinnvoll, ein zu untersuchendes Dateisystem:

read-only

einzubinden.

Midnight Commander kann dann zum Navigieren und Lesen verwendet werden, während Schreiboperationen auf diesem Mount vom Dateisystem beziehungsweise Mount blockiert werden.

Wichtige Grenze

Midnight Commander selbst macht ein Dateisystem nicht read-only.

Der Schutz muss auf der Ebene des:

Mounts,

Dateisystems,

Blockgeräts

oder geeigneter anderer Schutzmaßnahmen

realisiert werden.

Read-only und Sicherungsziel

Ein typischer sicherer Aufbau lautet:

Quellmedium

read-only eingebunden.

Zielmedium

read-write eingebunden.

Midnight Commander

linkes und rechtes Panel für Quelle und Ziel.

Geräteidentität zuerst prüfen

Vor einer größeren Kopieraktion muss eindeutig geklärt werden:

Welcher Mountpunkt gehört zum Quellmedium?

Welcher zum Zielmedium?

Ist auf dem Ziel genügend freier Speicher vorhanden?

Ist das Ziel tatsächlich beschreibbar?

Kein blindes Vertrauen in Pfadnamen

Ein Verzeichnisname wie:

`/mnt/backup`

beweist nicht, dass dort tatsächlich das gewünschte Sicherungsmedium eingehängt ist.

Vor kritischen Operationen muss die Mountsituation geprüft werden.

Freier Speicher

Vor umfangreichen Kopiervorgängen sollte die freie Kapazität des Ziels kontrolliert werden.

Ein während der Rettung volllaufendes Zielmedium kann einen Kopiervorgang unterbrechen und die Übersicht erschweren.

Interner Editor mcedit

Der mit Midnight Commander bereitgestellte Editor kann unabhängig vom Dateimanager gestartet werden:

`mcedit`

Version 4.8.33 besitzt hierfür eine eigene Programmdokumentation.

Syntaxhervorhebung

mcedit unterstützt Syntaxdefinitionen.

Systemweite Syntaxdateien liegen laut Debian-Dokumentation unter:

`/usr/share/mc/syntax/`.

Benutzerspezifische Editorinformationen liegen unter:

`~/.local/share/mc/mcedit/`.

Konfiguration

Die benutzerspezifische Midnight-Commander-Konfiguration befindet sich typischerweise unter:

`~/.config/mc/`.

Die Debian-Dokumentation nennt unter anderem:

`~/.config/mc/ini`

für die benutzerspezifische Konfiguration.

Rescue-Bedeutung der Benutzerkonfiguration

Änderungen an der MC-Konfiguration verändern grundsätzlich die Arbeitsumgebung von Midnight Commander.

Sie verändern dadurch nicht automatisch die untersuchten Quelldateien.

Davon zu unterscheiden sind Dateioperationen, die innerhalb von MC auf einem gemounteten Zielsystem ausgeführt werden.

Mausunterstützung

Midnight Commander unterstützt Mausbedienung im Terminal.

Mit:

`-d`

beziehungsweise:

`--nomouse`

kann die Mausunterstützung deaktiviert werden.

Schwarz-Weiß-Darstellung

Mit:

`mc -b`

beziehungsweise:

`mc --nocolor`

wird eine Schwarz-Weiß-Darstellung erzwungen.

Farbdarstellung

Mit:

`mc -c`

beziehungsweise:

```
mc --color
```

kann die Farbdarstellung erzwungen werden.

Version anzeigen

Mit:

```
mc -V
```

beziehungsweise:

```
mc --version
```

wird die Version ausgegeben. Die Programmdokumentation empfiehlt diese Angabe ausdrücklich auch bei Fehlerberichten.

Hilfe

Midnight Commander besitzt eine integrierte Hilfe.

Innerhalb des Programms ist sie über:

```
F1
```

erreichbar.

Typischer Rescue-Fall: Benutzerdateien sichern

1. Quellmedium möglichst read-only einbinden.

2. Sicherungsmedium read-write einbinden.

3. Geräte- und Mountpunktzuordnung prüfen.

4. Midnight Commander starten.

Beispiel:

```
mc /mnt/quelle /mnt/backup
```

5. Quelle und Ziel nochmals visuell kontrollieren.

6. benötigte Dateien markieren.

7. mit F5 kopieren.

8. Kopierergebnis kontrollieren.

9. Quelle unverändert lassen.

Typischer Rescue-Fall: Konfigurationsdatei untersuchen

1. Zum Zielsystem navigieren.

2. Datei markieren.

3. F3 verwenden.

4. Inhalt lesen.

5. Nur wenn eine Änderung ausdrücklich erforderlich ist, auf einer geeigneten Arbeitsumgebung F4 verwenden.

Typischer Rescue-Fall: Konfiguration reparieren

Soll beispielsweise eine beschädigte Textkonfiguration korrigiert werden:

1. Original sichern.

2. prüfen, ob das Zielsystem tatsächlich beschreibbar sein soll.

3. Datei gegebenenfalls auf Arbeitskopie bearbeiten.

4. Änderung mit mcedit durchführen.

5. Ergebnis prüfen.

6. ursprüngliche Datei nicht ohne Sicherung überschreiben.

Typischer Rescue-Fall: gerettete Dateien sortieren

Nach PhotoRec oder Foremost können große Mengen rekonstruierter Dateien entstehen.

Midnight Commander kann anschließend beim:

Navigieren,

Anzeigen,

Umbenennen,

Sortieren

und Kopieren

dieser Ergebnisse helfen.

Wichtige Grenze

Midnight Commander verbessert nicht die Qualität einer PhotoRec- oder Foremost-Rekonstruktion.

Er verwaltet lediglich die bereits vorhandenen Ergebnisse.

Typischer Rescue-Fall: Logs sichern

Ein sinnvolles Vorgehen kann sein:

1. Quellsystem read-only einbinden.

2. Logverzeichnis im einen Panel öffnen.

3. Sicherungsverzeichnis im anderen Panel öffnen.

4. relevante Logs mit F5 kopieren.

5. Kopien anschließend analysieren.

Typischer Rescue-Fall: Remote-Dateien

Wenn ein Remote-Dateisystem beispielsweise mit:

A.28 SSHFS

in das lokale Dateisystem eingebunden wurde, kann Midnight Commander den Mountpunkt wie andere zugängliche Verzeichnisse verwenden.

Sicherheitsgrenze

Midnight Commander übernimmt dabei die Eigenschaften des zugrunde liegenden Mounts.

Ein read-only SSHFS-Mount bleibt für normale Dateioperationen read-only.

Ein read-write eingebundener Remote-Mount kann dagegen durch Dateioperationen verändert werden.

Midnight Commander und SSH

Midnight Commander kann in einem Terminal auch über eine SSH-Verbindung eingesetzt werden.

Das Projekt beschreibt ausdrücklich, dass die textbasierte Oberfläche auf einer normalen Konsole, in grafischen Terminals und über SSH beziehungsweise andere Remote-Shells verwendet werden kann.

Midnight Commander und GNU ddrescue

Die Rollen sind grundsätzlich verschieden:

GNU ddrescue

Blockebene und Acquisition problematischer Medien.

Midnight Commander

Datei- und Verzeichnisebene.

Ein typischer Ablauf kann deshalb lauten:

beschädigter Datenträger

↓

ddrescue-Abbild

↓

Dateisystem beziehungsweise rekonstruierte Daten zugänglich machen

↓

Midnight Commander

Dateien sichten und auf ein Zielmedium organisieren.

Midnight Commander und TestDisk

TestDisk

rekonstruiert beziehungsweise untersucht Partitions- und bestimmte Dateisystemstrukturen.

Midnight Commander

verwaltet zugängliche Dateien.

Midnight Commander und PhotoRec / Foremost

PhotoRec / Foremost

rekonstruieren Dateien über File Carving.

Midnight Commander

hilft anschließend bei der Sichtung und Organisation der Ergebnisse.

Midnight Commander und Extundelete

Extundelete

versucht gelöschte Dateien aus ext-Dateisystemen wiederherzustellen.

Midnight Commander

verwaltet die wiederhergestellten Dateien anschließend auf Dateisystemebene.

Midnight Commander und GParted

GParted

arbeitet mit Partitionen und Dateisystemstrukturen.

Midnight Commander

arbeitet mit Dateien und Verzeichnissen innerhalb zugänglicher Dateisysteme.

Midnight Commander und GNOME Disks

GNOME Disks

verwaltet beziehungsweise untersucht Laufwerke, Partitionen und Images.

Midnight Commander

ist für die Dateioperationen innerhalb zugänglicher Verzeichnisstrukturen zuständig.

Midnight Commander und ClamAV

Dateien, die mit Midnight Commander lokalisiert oder gerettet wurden, können anschließend mit:

ClamAV

untersucht werden.

Bei potenziell kompromittierten Dateien gilt weiterhin:

nicht ausführen.

Midnight Commander und Binwalk / Hexedit

Binwalk

analysiert Binärdaten und eingebettete Strukturen.

Hexedit

arbeitet auf Byteebene.

Midnight Commander

dient dagegen primär der Navigation und Verwaltung der betreffenden Dateien.

Grenzen

Midnight Commander ist kein File-Carving-Werkzeug.

Midnight Commander ist kein Partitionsrettungsprogramm.

Midnight Commander ist kein Block-Imager.

Midnight Commander ist kein SMART-Diagnosewerkzeug.

Midnight Commander überprüft keine Dateisystemintegrität.

Midnight Commander garantiert keine forensische Metadatenerhaltung.

Midnight Commander macht eine Quelle nicht automatisch read-only.

Midnight Commander verhindert keine Bedienfehler.

Midnight Commander macht Shell-Kommandos nicht ungefährlich.

Midnight Commander erkennt nicht automatisch, ob Quelle und Ziel verwechselt wurden.

Sicherheitshinweise

Vor jeder schreibenden Operation das aktive Panel prüfen.

Quelle und Ziel eindeutig identifizieren.

Bei Rescue-Arbeiten **F5 Kopieren** grundsätzlich gegenüber **F6 Verschieben** bevorzugen, wenn Originaldaten erhalten bleiben sollen.

F8 Löschen nur nach eindeutiger Kontrolle verwenden.

Originalquellen möglichst read-only einbinden.

Verdächtige ausführbare Dateien nicht mit Enter starten.

Zur ersten Untersuchung einer Datei F3 statt F4 verwenden.

Vor dem Bearbeiten wichtiger Konfigurationen eine Sicherung anlegen.

Nicht davon ausgehen, dass eine normale Dateikopie ein forensisches Abbild darstellt.

Bei physisch instabilen Datenträgern nicht große Dateibäume wiederholt mit einem Dateimanager durchsuchen, wenn zunächst ein ddrescue-Abbild erforderlich ist.

Vor großen Kopiervorgängen freien Speicherplatz und Mountpunkte kontrollieren.

Technische Einordnung in JUMBO BLOCK Rescue

Midnight Commander bildet innerhalb von Block A den eigenständigen Bereich:

Dateien.

Seine Rolle unterscheidet sich deutlich von den vorherigen Diagnose- und Recovery-Werkzeugen.

Ein vollständiger Rescue-Workflow kann beispielsweise lauten:

Hardware identifizieren

HWInfo / Inxi / LSHW / dmidecode

↓

Datenträgerzustand beurteilen

SMART / NVMe

↓

instabiles Medium sichern

GNU ddrescue

↓

Partitions- oder Dateisystemstrukturen untersuchen

TestDisk / Extundelete

↓

Dateien rekonstruieren, falls erforderlich

PhotoRec / Foremost

↓

Dateien verwalten und sichern

Midnight Commander

Damit übernimmt MC die praktische Datei- und Verzeichnisarbeit innerhalb einer bereits zugänglichen Rescue-Umgebung.

Seine Zwei-Panel-Struktur eignet sich besonders für das klare Gegenüberstellen von:

Quelle

und:

Sicherungsziel.

Gerade diese Stärke bringt aber auch Verantwortung mit sich:

Ein falsches Panel, ein falsch identifizierter Mountpunkt oder die Verwechslung von F5, F6 und F8 kann Originaldaten verändern oder löschen.

Der zentrale Rescue-Grundsatz für Midnight Commander lautet deshalb:

Erst Quelle und Ziel eindeutig bestimmen – dann kopieren. Verschieben, Bearbeiten oder Löschen nur, wenn die Veränderung ausdrücklich beabsichtigt ist.

A.35 QPDF

Bereich: Dokumente
Programm: QPDF
Baseline 20261001: QPDF 12.2.0
Installiertes Paket: `qpdf 12.2.0-1`
Zugehörige Bibliothek: `libqpdf30 12.2.0-1`
Bedienung: Kommandozeile
Aufgabe: PDF-Dateien strukturell prüfen, reparierend neu schreiben und gezielt transformieren

Zweck

QPDF ist ein Kommandozeilenwerkzeug zur technischen Verarbeitung von PDF-Dateien.

Im JUMBO BLOCK Rescue System liegt der Schwerpunkt auf:

Prüfung problematischer oder geretteter PDF-Dateien,

Erkennung struktureller PDF-Fehler,

Neuschreiben beschädigter PDF-Strukturen,

Prüfung und Verarbeitung verschlüsselter PDFs,

Auswahl und Zusammenstellung einzelner Seiten

sowie:

Aufteilung größerer PDF-Dokumente.

QPDF ist kein PDF-Viewer und kein klassischer Dokumenteditor. Es bearbeitet die interne Struktur einer PDF-Datei.

Einsatzgebiet im Rescue-System

QPDF kommt typischerweise zum Einsatz, nachdem eine PDF-Datei bereits von einem Datenträger, Dateisystem oder Rescue-Image gesichert beziehungsweise rekonstruiert wurde.

Ein typischer Ablauf kann sein:

Datenträger retten

↓

Datei wiederherstellen

↓

PDF mit QPDF technisch prüfen

↓

gegebenenfalls reparierte Arbeitskopie erzeugen

↓

Ergebnis mit einem PDF-Reader visuell kontrollieren

PDF prüfen

Der wichtigste Diagnosebefehl lautet:

```
qpdf --check DATEI.pdf
```

QPDF untersucht damit die interne PDF-Struktur und meldet erkannte Fehler oder Warnungen.

Das ist besonders sinnvoll, wenn eine PDF:

nicht geöffnet werden kann,

nach einer Datenrettung verdächtig erscheint,

von verschiedenen PDF-Readern unterschiedlich verarbeitet wird

oder:

beim Öffnen Fehlermeldungen erzeugt.

Warnungen richtig bewerten

Eine Warnung bedeutet nicht automatisch, dass das Dokument unbrauchbar ist.

QPDF besitzt Recovery-Mechanismen und kann bestimmte beschädigte PDF-Strukturen beim Einlesen rekonstruieren.

Umgekehrt beweist eine erfolgreiche strukturelle Prüfung nicht, dass jede Seite optisch und inhaltlich vollständig ist.

Deshalb gilt:

Technische Prüfung mit QPDF und visuelle Prüfung mit einem PDF-Reader ergänzen sich.

Beschädigte PDF neu schreiben

Kann QPDF eine beschädigte Datei noch lesen, kann sie als neue PDF-Datei geschrieben werden.

Beispiel:

```
qpdf problem.pdf recovered.pdf
```

Dabei wird die von QPDF interpretierte Dokumentstruktur neu ausgegeben.

Anschließend sollte die neue Datei erneut geprüft werden:

```
qpdf --check recovered.pdf
```

Original erhalten

Die beschädigte Originaldatei darf bei einem Rescue-Vorgang nicht unnötig überschrieben werden.

Empfohlen ist:

`problem.pdf`

→ unverändertes Original

`recovered.pdf`

→ Arbeits- beziehungsweise Reparaturfassung.

Der zentrale Grundsatz lautet:

Original erhalten – Reparaturversuch in eine neue Datei schreiben.

Grenzen der Reparatur

QPDF kann nur Informationen verarbeiten, die in der geretteten Datei noch vorhanden beziehungsweise aus deren Struktur rekonstruierbar sind.

Fehlen beispielsweise nach einer unvollständigen Datenrettung tatsächliche Dateibereiche, kann QPDF diese Inhalte nicht neu erzeugen.

QPDF repariert deshalb nicht jede beschädigte PDF.

Seitenzahl prüfen

Mit:

```
qpdf --show-npages DATEI.pdf
```

kann die Anzahl der Seiten ausgegeben werden.

Dies ist beispielsweise nach einer Wiederherstellung hilfreich, um die rekonstruierte Datei mit der erwarteten Dokumentgröße zu vergleichen.

Die Seitenzahl allein beweist jedoch nicht die Vollständigkeit der Seiteninhalte.

Seiten auswählen und zusammenstellen

QPDF kann mit:

`--pages`

bestimmte Seiten auswählen, neu anordnen und Seiten aus mehreren PDF-Dateien zu einem neuen Dokument zusammenstellen.

Dies kann im Rescue-Betrieb sinnvoll sein, wenn:

nur bestimmte Seiten benötigt werden,

einzelne Seiten eines Dokuments gerettet werden sollen

oder:

verwertbare Seiten aus mehreren Dateien in einer neuen PDF zusammengeführt werden sollen.

Das Ergebnis sollte grundsätzlich in eine neue Datei geschrieben werden.

PDF aufteilen

Mit:

`--split-pages`

kann eine PDF in mehrere Dateien zerlegt werden.

Dies kann beispielsweise hilfreich sein, wenn ein großes Dokument in kleinere Einheiten aufgeteilt oder ein Problem auf bestimmte Seitenbereiche eingegrenzt werden soll.

Verschlüsselte PDFs

QPDF kann erkennen und anzeigen, ob beziehungsweise wie eine PDF verschlüsselt ist.

Dazu stehen unter anderem Funktionen zur:

Prüfung des Verschlüsselungsstatus,

Anzeige von Verschlüsselungsinformationen

und:

Verarbeitung mit vorhandenen Zugangsdaten

zur Verfügung.

Entschlüsseln

Mit:

`--decrypt`

kann bei berechtigtem Zugriff eine unverschlüsselte Arbeitskopie erzeugt werden.

Dabei ist zu beachten:

Die neue Datei besitzt anschließend nicht mehr den bisherigen PDF-Verschlüsselungsschutz.

Sie muss deshalb entsprechend sicher gespeichert werden.

Kein Passwortknacker

QPDF ist kein Werkzeug zum Brute-Force-Knacken unbekannter PDF-Passwörter.

Im Rescue-System wird es nur mit vorhandener Berechtigung und erforderlichen Zugangsdaten eingesetzt.

Eingebettete Dateien

PDF-Dokumente können Dateianhänge enthalten.

QPDF kann solche eingebetteten Dateien untersuchen und extrahieren.

Dies kann bei einer Dokumentrettung relevant sein, wenn eine PDF zusätzliche Dateien enthält.

Sicherheitsregel für Anhänge

Extrahierte unbekannte Dateien dürfen nicht automatisch geöffnet oder ausgeführt werden.

Bei verdächtiger oder unbekannter Herkunft können sie zusätzlich mit:

ClamAV

untersucht werden.

Digitale Signaturen

Das Neuschreiben einer PDF verändert deren Dateistruktur und normalerweise auch ihren Hashwert.

Bei digital signierten Dokumenten kann dies Auswirkungen auf die Signaturvalidierung haben.

Deshalb gilt:

Digital signierte Original-PDFs immer unverändert erhalten.

Analyse oder Reparatur ausschließlich an einer Kopie durchführen.

QPDF ersetzt keine eigenständige kryptografische Signaturprüfung.

Typischer Rescue-Fall: PDF lässt sich nicht öffnen

1. Original sichern.

2. PDF prüfen:

```
qpdf --check problem.pdf
```

3. Warnungen und Fehler dokumentieren.

4. Wenn QPDF die Datei lesen kann, neue Arbeitskopie erzeugen:

```
qpdf problem.pdf recovered.pdf
```

5. Arbeitskopie erneut prüfen:

```
qpdf --check recovered.pdf
```

6. recovered.pdf mit einem PDF-Reader visuell kontrollieren.

7. Original unverändert aufbewahren.

Typischer Rescue-Fall: PDF aus PhotoRec oder Foremost

Eine rekonstruierte PDF kann zunächst mit:

```
qpdf --check DATEI.pdf
```

untersucht werden.

Anschließend sollte:

die Seitenzahl geprüft,

das Dokument visuell kontrolliert

und bei strukturellen Problemen gegebenenfalls eine neu geschriebene Arbeitskopie erzeugt werden.

QPDF kann dabei nur mit den Daten arbeiten, die PhotoRec beziehungsweise Foremost tatsächlich wiederhergestellt haben.

Typischer Rescue-Fall: einzelne Seiten retten

Wenn nur bestimmte Seiten eines noch verarbeitbaren Dokuments benötigt werden:

Original erhalten.

↓

Benötigte Seiten mit `--pages` auswählen.

↓

Neue PDF erzeugen.

↓

Neue Datei mit `--check` prüfen.

↓

Ergebnis visuell kontrollieren.

Typischer Rescue-Fall: verschlüsselte PDF

Verschlüsselungsstatus feststellen.

↓

Vorhandene Zugangsdaten verwenden.

↓

Falls erforderlich entschlüsselte Arbeitskopie erzeugen.

↓

Arbeitskopie angemessen schützen.

Das verschlüsselte Original bleibt erhalten.

QPDF und andere Rescue-Werkzeuge

GNU ddrescue

rettet Daten auf Blockebene.

TestDisk / Extundelete

arbeiten an Partitions- beziehungsweise Dateisystemstrukturen und Wiederherstellung.

PhotoRec / Foremost

rekonstruieren Dateien über File Carving.

Midnight Commander

verwaltet gerettete Dateien und Verzeichnisse.

QPDF

untersucht anschließend die interne Struktur einer PDF-Datei.

Damit liegt QPDF typischerweise am Ende der technischen Rescue-Kette.

Grenzen

QPDF ist kein PDF-Viewer.

QPDF ist kein OCR-System.

QPDF ist kein klassischer PDF-Inhaltseditor.

QPDF kann fehlende Dateidaten nicht neu erzeugen.

QPDF repariert nicht jede beschädigte PDF.

QPDF prüft nicht automatisch die visuelle Vollständigkeit aller Seiten.

QPDF ist kein Passwort-Brute-Force-Werkzeug.

QPDF ist kein Antivirenprogramm.

QPDF ersetzt keine kryptografische Signaturprüfung.

Ein neu geschriebenes PDF ist nicht byteidentisch mit dem Original.

Sicherheitshinweise

Originaldateien vor jeder Transformation sichern.

Beschädigte PDFs zunächst mit `--check` untersuchen.

Reparaturversuche in eine neue Datei schreiben.

Original und Arbeitskopie eindeutig voneinander trennen.

Digital signierte Originale unverändert erhalten.

Entschlüsselte Arbeitskopien entsprechend schützen.

Unbekannte eingebettete Dateien nicht automatisch ausführen.

Nach einer strukturellen Reparatur das Dokument zusätzlich visuell kontrollieren.

Eine erfolgreiche QPDF-Prüfung nicht mit einer Sicherheitsprüfung des Dokuments verwechseln.

Technische Einordnung in JUMBO BLOCK Rescue

QPDF bildet den Bereich:

Dokumente

und damit den Abschluss von Block A.

Ein typischer Rescue-Weg kann lauten:

Datenträger sichern

↓

Dateisystem beziehungsweise Dateien wiederherstellen

↓

gerettete Dateien mit Midnight Commander verwalten

↓

problematische PDF mit QPDF prüfen

↓

gegebenenfalls reparierte Arbeitskopie erzeugen

↓

Ergebnis visuell kontrollieren

QPDF übernimmt damit eine klar abgegrenzte Aufgabe:

PDF-Dokumente nach ihrer Rettung technisch prüfen und ihre interne Struktur, soweit möglich, in einer neuen Arbeitskopie wieder konsistent ausgeben.

Der zentrale Rescue-Grundsatz lautet:

Original erhalten – PDF-Struktur prüfen – Reparatur nur an einer Arbeitskopie – Ergebnis anschließend technisch und visuell kontrollieren.

Block B

JUMBO BLOCK Rescue basiert technisch auf MX Linux, ist jedoch als **betriebssystemübergreifendes Rescue-System** konzipiert.

Die MX-Basis stellt zahlreiche System- und Verwaltungswerkzeuge bereit. Für dieses Handbuch werden jedoch nur diejenigen MX-Werkzeuge einzeln beschrieben, die einen konkreten Nutzen für Rescue-Aufgaben, die Arbeit mit bootfähigen Medien oder die sichere Überführung und Prüfung von Systemständen und ISO-Abbildern besitzen.

Reine Verwaltungs- und Komfortfunktionen des MX-Basissystems sind nicht Gegenstand der ausführlichen Rescue-Dokumentation.

Werkzeuggruppen

Sechs relevante MX-Werkzeuge:

B.1 – MX Boot Repair
Reparatur von GRUB-basierten Linux-Startumgebungen sowie Neuerzeugung von GRUB-Konfiguration und initramfs.

B.2 – UEFI Manager
Prüfung und Verwaltung von UEFI-Boot-Einträgen und Bootreihenfolgen zur Diagnose und Korrektur von Startproblemen.

B.3 – Chroot Rescue Scan
Erkennung installierter Linux-Systeme und Bereitstellung einer chroot-Umgebung für gezielte Reparaturarbeiten an nicht mehr regulär startfähigen Installationen.

B.4 – MX Live USB Maker
Erstellung und Vervielfältigung bootfähiger USB-Medien aus ISO-Abbildern oder vorhandenen Live-Systemen.

B.5 – MX Snapshot
Erstellung bootfähiger ISO-Abbilder aus einem vorbereiteten Systemstand. Ausgewählte gerettete oder zu untersuchende Dateien können kontrolliert in einen solchen Snapshot übernommen und gemeinsam mit dem Rescue-System gesichert beziehungsweise überführt werden.

B.6 – ISO Signature Verifier
Betriebssystemunabhängige Integritäts- und Authentizitätsprüfung von ISO-Abbildern anhand geeigneter Signatur- und Prüfsummeninformationen.

Die sechs Werkzeuge ergänzen damit die spezialisierten Programme aus **Block A – JB-Rescue**, ohne den Schwerpunkt des Handbuchs auf die zugrunde liegende MX-Linux-Distribution zu verlagern.

JUMBO BLOCK Rescue nutzt MX Linux als technische Basis – sein Einsatzgebiet ist jedoch nicht auf MX Linux oder Linux-Systeme beschränkt.

Zuordnung der Bereiche, Programme und Versionen

Nr.	Bereich	Programm	Baseline-Version
B.1	Boot-Reparatur	MX Boot Repair	26.06.1 ff.
B.2	Boot / UEFI	UEFI Manager	26.07.1 ff.
B.3	Linux-Systemrettung	Chroot Rescue Scan	0.2.6+5 ff.
B.4	Rescue-Medium	MX Live USB Maker	26.04.1 ff.
B.5	Systemabbild / ISO-Erstellung	MX Snapshot	26.08 ff.
B.6	Integrität / Authentizität	ISO Signature Verifier	2026.09.01 ff.

B.1 MX Boot Repair

Bereich: Boot-Reparatur
Programm: MX Boot Repair
Baseline 20261001: 26.06.1
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Reparatur von GRUB-basierten Linux-Startumgebungen

Zweck

MX Boot Repair ist ein grafisches Werkzeug zur Reparatur bestimmter Bootprobleme von Linux-Systemen.

Im JUMBO BLOCK Rescue System ist das Programm insbesondere für installierte Linux-Systeme relevant, die **GRUB** als Bootloader verwenden.

Die wesentlichen Funktionen sind:

- GRUB-Konfigurationsdatei neu erzeugen,
- GRUB-Bootloader neu installieren,
- beschädigte oder fehlende initramfs-Abbilder neu erzeugen.

Bedeutung für JUMBO BLOCK Rescue

JUMBO BLOCK Rescue ist nicht auf die Reparatur von MX Linux beschränkt.

MX Boot Repair ist deshalb als **spezialisiertes Linux-/GRUB-Werkzeug innerhalb des universellen Rescue-Systems** einzuordnen.

Es kann hilfreich sein, wenn ein installiertes Linux-System wegen einer beschädigten oder fehlerhaften GRUB-Konfiguration nicht mehr startet.

Es ist dagegen **kein universelles Reparaturwerkzeug für die Bootloader aller Betriebssysteme**.

GRUB-Konfiguration reparieren

Die Funktion:

Repair GRUB configuration file

erzeugt die GRUB-Konfiguration des ausgewählten Linux-Systems neu.

MX empfiehlt diese Funktion als ersten Reparaturversuch bei entsprechenden GRUB-Problemen.

Ein mögliches Fehlerbild ist beispielsweise:

```
grub rescue>
```

Die Funktion setzt voraus, dass GRUB selbst grundsätzlich an der vorgesehenen Stelle installiert ist.

Ist der eigentliche Bootloader beschädigt oder nicht mehr vorhanden, kann zunächst eine Neuinstallation von GRUB erforderlich sein.

GRUB-Bootloader neu installieren

MX Boot Repair kann GRUB neu installieren.

Abhängig vom verwendeten Startverfahren sind dafür unterschiedliche Ziele vorgesehen:

ESP

EFI System Partition bei UEFI-Systemen.

MBR

Master Boot Record bei klassischen BIOS-/Legacy-Systemen.

PBR

Partition Boot Record beziehungsweise Bootsektor einer Partition.

Nach einer Neuinstallation von GRUB empfiehlt die MX-Dokumentation, anschließend auch die GRUB-Konfiguration neu erzeugen zu lassen.

UEFI-Systeme

Für eine UEFI-Reparatur mit MX Boot Repair muss das Rescue-System selbst im **UEFI-Modus** gestartet worden sein.

Bei der Reparatur müssen insbesondere:

- die richtige EFI-Systempartition,
- die richtige Linux-Root-Partition

und gegebenenfalls:

- eine separate `/boot`-Partition

eindeutig identifiziert werden.

Die automatische Vorauswahl des Programms darf bei komplexen Installationen nicht ungeprüft übernommen werden. Darauf weist die MX-Dokumentation ausdrücklich hin.

Legacy-/MBR-Systeme

Bei klassischen BIOS-/Legacy-Installationen kann GRUB in den MBR des betreffenden Datenträgers neu installiert werden.

Auch hier muss das Zielgerät eindeutig identifiziert werden.

Für eine Legacy-Reparatur muss JUMBO BLOCK Rescue entsprechend im Legacy-/MBR-Modus gestartet worden sein.

Initramfs neu erzeugen

MX Boot Repair bietet außerdem:

Regenerate initramfs images

Damit können fehlende oder beschädigte initramfs-Abbilder im `/boot`-Bereich eines Linux-Systems neu erzeugt werden.

Das initramfs stellt beim Linux-Start die frühe Benutzerumgebung bereit, in der unter anderem das eigentliche Root-Dateisystem gefunden und vorbereitet wird.

Ein beschädigtes oder fehlendes initramfs kann deshalb verhindern, dass ein Linux-System vollständig startet.

Typischer Rescue-Ablauf

Bei einem nicht mehr startenden GRUB-basierten Linux-System:

1. JUMBO BLOCK Rescue starten.

2. Datenträger und Partitionen eindeutig identifizieren.

3. Prüfen, ob das Zielsystem UEFI oder Legacy/BIOS verwendet.

4. Root-, EFI- und gegebenenfalls `/boot`-Partition eindeutig bestimmen.

5. Zunächst die GRUB-Konfiguration reparieren, sofern das Fehlerbild dazu passt.

6. Falls erforderlich GRUB neu installieren.

7. Anschließend die GRUB-Konfiguration erneut erzeugen.

8. Bei beschädigtem oder fehlendem initramfs dieses neu erzeugen.

9. System neu starten und Bootvorgang kontrollieren.

Dual- und Multi-Boot-Systeme

Besondere Vorsicht ist bei Systemen erforderlich, auf denen mehrere Betriebssysteme installiert sind.

Eine EFI-Systempartition kann Bootdateien mehrerer Betriebssysteme enthalten.

Außerdem kann sich lediglich die UEFI-Bootreihenfolge verändert haben, obwohl die eigentlichen Bootloader weiterhin vorhanden sind.

Die MX-Dokumentation nennt ausdrücklich den Fall, dass Windows-Updates den Windows Boot Manager in der UEFI-Bootreihenfolge vor einen vorhandenen Linux-Eintrag setzen können. In einem solchen Fall sollte zunächst die UEFI-Konfiguration geprüft werden, bevor GRUB neu installiert wird.

Für diese Aufgabe steht in JUMBO BLOCK Rescue **B.2 – UEFI Manager** zur Verfügung.

Grenzen

MX Boot Repair ist auf **Linux-/GRUB-bezogene Reparaturaufgaben** ausgerichtet.

Es ist insbesondere:

kein allgemeines Windows-Boot-Reparaturwerkzeug,

kein macOS-Boot-Reparaturwerkzeug,

kein Dateisystem-Reparaturprogramm,

kein Partitionsrettungsprogramm

und:

kein Datenrettungsprogramm.

Ein Windows-System auf demselben Rechner kann für die Beurteilung der Bootkonfiguration relevant sein. Daraus folgt jedoch nicht, dass MX Boot Repair den Windows-Bootloader selbst repariert.

Sicherheitshinweise

Bootloader-Reparaturen sind schreibende Eingriffe in die Startkonfiguration eines Datenträgers.

Vor der Ausführung müssen deshalb:

Zieldatenträger, Root-Partition, EFI-Systempartition und Bootverfahren eindeutig identifiziert sein.

Bei einem physisch instabilen oder beschädigten Datenträger hat zunächst die Datensicherung beziehungsweise Image-Erstellung Vorrang.

Bei Dual- und Multi-Boot-Systemen muss berücksichtigt werden, dass mehrere Betriebssysteme dieselbe EFI-Systempartition verwenden können.

Eine bloß falsche UEFI-Bootreihenfolge sollte nicht durch eine unnötige Neuinstallation des Bootloaders „repariert“ werden.

Automatisch vorgeschlagene Zielpartitionen sind vor dem Schreiben zu kontrollieren.

Zusammenspiel mit anderen Rescue-Werkzeugen

UEFI Manager

prüft und verwaltet UEFI-Boot-Einträge und Bootreihenfolge.

Chroot Rescue Scan

ermöglicht den Zugriff auf installierte Linux-Systeme über eine chroot-Umgebung, wenn weitergehende Reparaturen erforderlich sind.

GParted / GNOME Disks

helfen bei der Identifikation von Datenträgern und Partitionen.

SMART Diagnose / NVMe Diagnose

prüfen den Zustand des zugrunde liegenden Datenträgers.

GNU ddrescue

sichert einen instabilen oder beschädigten Datenträger, bevor weitere Reparaturmaßnahmen durchgeführt werden.

TestDisk

ist für beschädigte beziehungsweise verlorene Partitionsstrukturen zuständig und erfüllt damit eine andere Aufgabe als MX Boot Repair.

Technische Einordnung in JUMBO BLOCK Rescue

MX Boot Repair gehört zu:

Block B – MX-Werkzeuge → Boot-Reparatur

Seine Bedeutung innerhalb von JUMBO BLOCK Rescue ist klar begrenzt:

Es erweitert das universelle Rescue-System um eine grafische Reparaturmöglichkeit für GRUB-basierte Linux-Installationen.

Andere Betriebssysteme und andere Bootloader benötigen gegebenenfalls eigene Reparaturverfahren.

Der zentrale Rescue-Grundsatz lautet:

Zuerst Bootverfahren und Zielpartitionen eindeutig identifizieren – nur tatsächlich beschädigte Komponenten reparieren – bei instabilen Datenträgern vor jeder Bootloader-Reparatur die Daten sichern.

Offizielle MX-Dokumentation: MX Boot Repair

B.2 UEFI Manager

Bereich: Boot / UEFI
Programm: UEFI Manager
Baseline 20261001: 26.07.1
Bedienung: Grafische Benutzeroberfläche
Aufgabe: UEFI-Boot-Einträge anzeigen und verwalten

Zweck

UEFI Manager ist ein grafisches Werkzeug zur Verwaltung der im UEFI-Firmwarebereich eines Rechners gespeicherten Boot-Einträge.

Für JUMBO BLOCK Rescue ist insbesondere die Funktion **Manage UEFI Entries** relevant. Damit lässt sich feststellen, welche Startoptionen vorhanden sind und in welcher Reihenfolge sie verwendet werden.

Einsatzgebiet im Rescue-System

UEFI Manager kann bei einem vermeintlichen Bootproblem helfen, bevor tatsächlich ein Bootloader repariert oder neu installiert wird.

Ein Betriebssystem kann weiterhin vollständig vorhanden und sein Bootloader intakt sein, obwohl der Rechner den falschen UEFI-Eintrag bevorzugt.

Die MX-Dokumentation zu Boot Repair empfiehlt deshalb ausdrücklich, bei entsprechenden Dual-Boot-Problemen zunächst die UEFI-Konfiguration zu überprüfen.

Voraussetzung

UEFI Manager benötigt ein UEFI-System und muss in einer Umgebung ausgeführt werden, die selbst im **UEFI-Modus** gestartet wurde.

Die MX-Dokumentation nennt als Kontrollmöglichkeit das Vorhandensein von:

`/sys/firmware/efi/efivars`

Außerdem muss eine zugängliche EFI-Systempartition vorhanden sein.

Vorhandene Boot-Einträge anzeigen

UEFI Manager zeigt die vorhandenen UEFI-Boot-Einträge in ihrer aktuellen Reihenfolge an.

Dabei können unter anderem:

- aktive und inaktive Einträge,
- der aktuell verwendete Boot-Eintrag,
- die Bootreihenfolge,
- ein gegebenenfalls gesetzter einmaliger nächster Boot-Eintrag

erkannt werden.

Damit lässt sich zunächst feststellen, ob ein benötigter Boot-Eintrag überhaupt noch vorhanden ist.

Bootreihenfolge ändern

Vorhandene Einträge können innerhalb der Bootreihenfolge nach oben oder unten verschoben werden.

Das ist beispielsweise relevant, wenn nach einer Änderung am Rechner plötzlich ein anderer Bootloader bevorzugt gestartet wird.

Eine falsche Bootreihenfolge ist von einem tatsächlich beschädigten Bootloader zu unterscheiden.

Boot Next

Mit **Boot Next** kann ein bestimmter UEFI-Eintrag ausschließlich für den nächsten Neustart ausgewählt werden.

Danach gilt wieder die normale Bootreihenfolge.

Für Rescue-Arbeiten ist dies besonders nützlich:

Ein vorhandener Boot-Eintrag kann zunächst getestet werden, **ohne sofort die permanente Bootreihenfolge zu verändern**.

Einträge aktivieren und deaktivieren

Vorhandene UEFI-Einträge können aktiviert beziehungsweise deaktiviert werden.

Ein deaktivierter Eintrag bleibt dabei vorhanden, wird aber nicht als aktiver Boot-Eintrag verwendet.

Für Diagnosezwecke ist das Deaktivieren deshalb grundsätzlich weniger endgültig als das Löschen eines Eintrags.

Einträge hinzufügen und entfernen

UEFI Manager kann außerdem neue Boot-Einträge zu vorhandenen EFI-Dateien anlegen und bestehende Einträge entfernen.

Das Entfernen eines Eintrags ist eine schreibende und potenziell folgenreiche Änderung.

Einträge sollten deshalb nicht gelöscht werden, nur weil ihre Funktion zunächst unbekannt ist.

Insbesondere bei Multi-Boot-Systemen können mehrere Einträge zu unterschiedlichen Betriebssystemen oder Wiederherstellungsumgebungen gehören.

Typisches Rescue-Szenario: falsches Betriebssystem startet

Ein Rechner besitzt mehrere Betriebssysteme oder Bootloader, startet aber plötzlich nicht mehr über den erwarteten Eintrag.

Vorgehen:

1. JUMBO BLOCK Rescue im UEFI-Modus starten.

2. UEFI Manager öffnen.

3. Vorhandene Boot-Einträge und ihre Reihenfolge prüfen.

4. Feststellen, ob der erwartete Eintrag noch vorhanden und aktiv ist.

5. Wenn möglich zunächst mit Boot Next testen.

6. Erst nach erfolgreicher Prüfung gegebenenfalls die dauerhafte Bootreihenfolge ändern.

Damit kann eine unnötige Neuinstallation eines funktionierenden Bootloaders vermieden werden.

Typisches Rescue-Szenario: Dual-Boot

Bei einem System mit beispielsweise Windows und Linux können mehrere UEFI-Einträge vorhanden sein.

Die MX-Dokumentation weist darauf hin, dass Windows-Updates die Bootreihenfolge verändern und den Windows Boot Manager vor einen vorhandenen Linux-Eintrag setzen können. Deshalb soll bei einem entsprechenden Fehlerbild zunächst die UEFI-Reihenfolge kontrolliert werden.

Das bedeutet nicht automatisch, dass GRUB oder ein anderer Bootloader beschädigt wurde.

EFI Stub Installer

UEFI Manager enthält zusätzlich einen **EFI Stub Installer** für Linux-Systeme. Damit kann ein Linux-Kernel direkt über einen UEFI-Eintrag gestartet werden, ohne einen Bootloader wie GRUB dazwischenzuschalten.

Für den allgemeinen Einsatz von JUMBO BLOCK Rescue ist dies eine Spezialfunktion und keine reguläre Rescue-Maßnahme.

Frugal EFI Stub Installer

Das Programm enthält außerdem einen speziellen EFI-Stub-Installer für frugale MX-/antiX-Installationen.

Diese Funktion betrifft die MX-/antiX-Infrastruktur und besitzt für die universelle Betriebssystemrettung von JUMBO BLOCK Rescue keine zentrale Bedeutung.

Sie wird deshalb hier nicht weiter behandelt.

Grenzen

UEFI Manager:

- repariert keine Dateisysteme,
- stellt keine Partitionen wieder her,
- repariert nicht den Inhalt eines beschädigten Bootloaders,

- ersetzt keine Datensicherung,
- ist für UEFI-Systeme bestimmt und nicht für klassische BIOS-/Legacy-Bootkonfigurationen.

Das Werkzeug verwaltet die **UEFI-Startkonfiguration**.

Ist beispielsweise GRUB selbst beschädigt, ist anschließend ein Werkzeug wie **MX Boot Repair** erforderlich.

Sicherheitshinweise

Vor Änderungen die vorhandene Bootreihenfolge dokumentieren.

Unbekannte Einträge nicht vorschnell löschen.

Bei Multi-Boot-Systemen besonders auf Einträge anderer Betriebssysteme achten.

Wenn möglich zunächst **Boot Next** verwenden, bevor die dauerhafte Reihenfolge verändert wird.

Eine vorhandene EFI-Datei oder ein vorhandener Boot-Eintrag beweist nicht automatisch, dass das zugehörige Betriebssystem vollständig funktionsfähig ist.

Das Löschen eines UEFI-Eintrags ist nicht dasselbe wie das Löschen des Betriebssystems, kann dessen regulären Start aber verhindern.

Technische Einordnung in JUMBO BLOCK Rescue

UEFI Manager gehört zu:

Block B – MX-Werkzeuge → Boot / UEFI

Im Gegensatz zu MX Boot Repair repariert das Werkzeug nicht primär den Linux-Bootloader.

Seine zentrale Rescue-Aufgabe lautet:

Vorhandene UEFI-Boot-Einträge untersuchen und eine fehlerhafte Bootauswahl beziehungsweise Bootreihenfolge erkennen und korrigieren.

Das macht UEFI Manager auch für Rechner mit unterschiedlichen Betriebssystemen relevant, weil die UEFI-Firmware oberhalb des jeweiligen Betriebssystems entscheidet, welcher konfigurierte EFI-Startweg verwendet wird.

Der zentrale Rescue-Grundsatz lautet:

Erst prüfen, ob lediglich der falsche UEFI-Eintrag gestartet wird – erst danach einen Bootloader reparieren oder neu installieren.

Offizielle MX-Dokumentation: UEFI Manager

B.3 Chroot Rescue Scan

Bereich: Linux-Systemrettung
Programm: Chroot Rescue Scan
Baseline 20261001: 0.2.6+5
Bedienung: Terminal / menügeführte Auswahl
Aufgabe: Installierte Linux-Systeme erkennen und für Reparaturarbeiten in einer chroot-Umgebung öffnen

Zweck

Chroot Rescue Scan ermöglicht den Zugriff auf ein installiertes Linux-System, obwohl dieses selbst nicht mehr regulär gestartet werden kann.

Das Werkzeug durchsucht vorhandene Partitionen nach Linux-Systemen, bindet erkannte Systeme ein und bietet anschließend die Auswahl eines Zielsystems für eine chroot-Sitzung an.

Damit können Reparaturbefehle innerhalb der Verzeichnis- und Systemumgebung der installierten Linux-Installation ausgeführt werden.

Bedeutung für JUMBO BLOCK Rescue

Chroot Rescue Scan ist eines der unmittelbar rescue-relevanten MX-Werkzeuge.

Es ist nicht auf die Reparatur der laufenden JUMBO BLOCK Rescue Umgebung ausgerichtet, sondern ermöglicht den Zugriff auf **andere installierte Linux-Systeme**.

Die MX-Dokumentation hebt insbesondere hervor, dass damit ein Zugang zu einem installierten System möglich sein kann, selbst wenn dessen `initrd.img` beschädigt ist. Außerdem können mehrere installierte Linux-Systeme untersucht werden, ohne für jedes System neu booten zu müssen.

Was ist eine chroot-Umgebung?

`chroot` verändert für einen Prozess und seine untergeordneten Prozesse das wahrgenommene Root-Verzeichnis `/`.

Wird beispielsweise die Root-Partition eines installierten Linux-Systems eingebunden und anschließend als chroot verwendet, erscheinen dessen Verzeichnisse:

`/etc`

`/usr`

`/var`

`/boot`

und weitere Systemverzeichnisse innerhalb der chroot wieder an ihren normalen Positionen.

Dadurch können viele administrative Reparaturbefehle so ausgeführt werden, als würde innerhalb des installierten Systems gearbeitet.

Automatische Suche

`chroot-rescue-scan` durchsucht die vorhandenen Partitionen nach Linux-Systemen.

Gefundene Systeme werden standardmäßig unter:

`/media`

eingebunden.

Anschließend stellt das Werkzeug ein Menü bereit, über das eines der gefundenen Linux-Systeme für die chroot-Sitzung ausgewählt werden kann.

Mehrere Linux-Installationen

Befinden sich mehrere Linux-Systeme auf einem Rechner, können diese erkannt und getrennt angeboten werden.

Das ist beispielsweise bei:

Dual-Boot-Systemen,

Multi-Boot-Systemen,

Testinstallationen

oder:

mehreren Linux-Installationen auf unterschiedlichen Datenträgern

hilfreich.

Vor Beginn einer Reparatur muss eindeutig festgestellt werden, **welches erkannte System tatsächlich bearbeitet werden soll**.

Vorbereitung der chroot-Umgebung

Das zugrunde liegende `chroot-rescue` bindet für das Zielsystem erforderliche Systembereiche wie:

`/sys`

`/proc`

und:

`/dev`

ein.

Für `/run` und `/tmp` werden temporäre Dateisysteme bereitgestellt. Anschließend wird in das ausgewählte Root-Verzeichnis gewechselt.

Damit erhält das installierte System während der Reparatur Zugriff auf wichtige Schnittstellen des laufenden Rescue-Kernels.

Shell im Zielsystem

Wenn kein bestimmter Befehl angegeben wird, versucht Chroot Rescue zunächst:

`/bin/bash`

und anschließend:

`/bin/sh`

im Zielsystem zu verwenden.

Nach erfolgreichem Wechsel befindet sich der Administrator in einer Shell, deren Root-Verzeichnis das installierte Linux-System ist.

Das ist ein entscheidender Punkt:

Befehle innerhalb der chroot können das installierte Zielsystem verändern.

Typische Rescue-Aufgaben

Innerhalb einer chroot-Umgebung können – abhängig von der jeweiligen Linux-Distribution und deren Werkzeugen – beispielsweise folgende Reparaturarbeiten durchgeführt werden:

- Konfigurationsdateien korrigieren,
- beschädigte oder fehlende Pakete reparieren,
- Boot-Konfigurationen bearbeiten,
- initramfs neu erzeugen,
- Benutzer- oder Berechtigungsprobleme korrigieren,
- fehlerhafte Einträge in `/etc/fstab` untersuchen,
- distributionsspezifische Wartungsbefehle ausführen.

Welche Befehle tatsächlich geeignet sind, richtet sich nach dem **installierten Zielsystem**.

Chroot Rescue Scan führt diese Reparaturen nicht automatisch durch. Es stellt die Umgebung bereit, in der sie durchgeführt werden können.

Netzwerkzugriff

Chroot Rescue kann versuchen, Netzwerkzugriff für die Zielumgebung bereitzustellen.

Die Werkzeugfamilie besitzt außerdem eine Option, mit der dieser Versuch unterdrückt werden kann.

Netzwerkzugriff ist beispielsweise erforderlich, wenn innerhalb der chroot Pakete aus den Repositories der installierten Distribution nachinstalliert oder repariert werden sollen.

Eine Netzwerkverbindung sollte jedoch nur hergestellt werden, wenn sie für die konkrete Reparatur erforderlich ist.

Beschädigtes initrd

Ein wesentlicher Rescue-Anwendungsfall ist ein installiertes Linux-System, das wegen eines beschädigten initrd beziehungsweise initramfs nicht mehr starten kann.

Da Chroot Rescue Scan das installierte System vom bereits laufenden JUMBO BLOCK Rescue System aus einbindet, muss das defekte Zielsystem nicht selbst erfolgreich booten.

Innerhalb der chroot können anschließend – sofern die jeweilige Distribution dies unterstützt und die übrige Installation ausreichend intakt ist – geeignete Werkzeuge zur Neuerzeugung des initramfs ausgeführt werden.

Typischer Rescue-Ablauf

1. JUMBO BLOCK Rescue starten.

2. Zustand des betroffenen Datenträgers prüfen.

Bei Hinweisen auf einen physisch instabilen Datenträger zunächst sichern beziehungsweise ein Image erstellen.

3. Chroot Rescue Scan starten.

4. Gefundene Linux-Systeme prüfen.

5. Das richtige Zielsystem eindeutig auswählen.

6. chroot-Umgebung öffnen.

7. Fehler innerhalb des installierten Systems diagnostizieren.

8. Nur die konkret erforderliche Reparatur durchführen.

9. chroot-Sitzung verlassen.

10. Eingebundene Dateisysteme sauber aushängen.

11. Zielsystem neu starten und Ergebnis prüfen.

Unterschied zu MX Boot Repair

MX Boot Repair

ist auf bestimmte Reparaturen von GRUB und initramfs spezialisiert und bietet dafür eine grafische Oberfläche.

Chroot Rescue Scan

stellt dagegen eine allgemeine administrative Arbeitsumgebung innerhalb eines installierten Linux-Systems bereit.

Dadurch können wesentlich weitergehende Reparaturen durchgeführt werden.

Diese größere Flexibilität bedeutet gleichzeitig eine größere Verantwortung: Innerhalb der chroot stehen je nach Zielsystem weitreichende administrative Möglichkeiten zur Verfügung.

Kein zweites gestartetes Betriebssystem

Eine chroot-Umgebung ist keine virtuelle Maschine und kein vollständig gestartetes zweites Betriebssystem.

Das Zielsystem verwendet weiterhin den **Kernel des laufenden JUMBO BLOCK Rescue Systems**.

Die chroot verändert im Wesentlichen die Sicht auf das Root-Dateisystem für die darin gestarteten Prozesse.

Deshalb können nicht alle Fehler eines installierten Systems innerhalb einer chroot realistisch reproduziert oder diagnostiziert werden.

Insbesondere kernelabhängige Probleme müssen entsprechend bewertet werden.

Grenzen gegenüber anderen Betriebssystemen

Chroot Rescue Scan ist ein Werkzeug für **Linux-Systeme**.

Es ist kein entsprechendes Reparaturverfahren für:

Windows,

macOS

oder andere Betriebssysteme, die keine geeignete Linux-chroot-Umgebung bereitstellen.

JUMBO BLOCK Rescue bleibt als Gesamtsystem universell einsetzbar; Chroot Rescue Scan stellt innerhalb dieses Systems lediglich eine spezialisierte Linux-Reparaturfunktion bereit.

Grenzen bei beschädigten Dateisystemen

Chroot Rescue Scan ersetzt keine Dateisystem- oder Datenträgerrettung.

Ist die Root-Partition:

physisch beschädigt,

nicht zuverlässig lesbar,

stark korrupt

oder:

gar nicht mehr vorhanden,

müssen zunächst die entsprechenden Rescue-Werkzeuge eingesetzt werden.

Ein chroot sollte nicht als erster Reparaturversuch auf einem physisch ausfallenden Datenträger verwendet werden.

Sicherheitshinweise

Vor dem Einstieg in die chroot das Zielsystem eindeutig identifizieren.

Bei mehreren Linux-Installationen niemals allein aufgrund der Reihenfolge im Auswahlmenü auf das richtige System schließen.

Bei einem instabilen Datenträger zuerst ein Image beziehungsweise eine Sicherung erstellen.

Innerhalb der chroot ausgeführte administrative Befehle können das Zielsystem dauerhaft verändern.

Paketverwaltung, Bootloader-Befehle, initramfs-Erzeugung und Änderungen unter `/etc` wirken auf die ausgewählte Installation.

Keine distributionsfremden Reparaturbefehle ungeprüft verwenden.

Nach Abschluss die chroot verlassen und eingebundene Dateisysteme sauber aushängen.

Technische Einordnung in JUMBO BLOCK Rescue

Chroot Rescue Scan gehört zu:

Block B – MX-Werkzeuge → Linux-Systemrettung

Es ergänzt JUMBO BLOCK Rescue um einen direkten administrativen Zugang zu nicht mehr startfähigen Linux-Installationen.

Seine Aufgabe ist dabei nicht die automatische Reparatur.

Die eigentliche Funktion lautet:

Installiertes Linux-System erkennen → einbinden → chroot-Umgebung vorbereiten → gezielte Reparatur innerhalb des Zielsystems ermöglichen.

Damit ist Chroot Rescue Scan deutlich universeller innerhalb der Linux-Welt als viele rein MX-spezifische Verwaltungswerkzeuge.

Der zentrale Rescue-Grundsatz lautet:

Zielsystem eindeutig identifizieren – bei gefährdeten Datenträgern zuerst sichern – chroot nur als Arbeitsumgebung verstehen – innerhalb des Zielsystems ausschließlich gezielte und nachvollziehbare Reparaturen durchführen.

Offizielle MX-Dokumentation: Chroot Rescue Scan

B.4 MX Live USB Maker

Bereich: Rescue-Medium
Programm: MX Live USB Maker
Baseline 20261001: 26.04.1
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Bootfähige Live-USB-Medien erstellen und vorhandene Live-Systeme klonen

Zweck

MX Live USB Maker erstellt bootfähige USB-Medien.

Als Quelle können unter anderem verwendet werden:

- eine ISO-Datei,
- ein bereits vorhandenes Live-USB-Medium,
- ein Live-CD/DVD-System,
- das aktuell laufende Live-System.

Damit dient das Werkzeug nicht unmittelbar der Reparatur eines Zielsystems, sondern der **Erstellung und Vervielfältigung bootfähiger Rescue-Medien**.

Bedeutung für JUMBO BLOCK Rescue

Innerhalb von JUMBO BLOCK Rescue kann MX Live USB Maker beispielsweise verwendet werden, um:

- eine ISO auf einen USB-Datenträger zu übertragen,
- ein vorhandenes Live-USB-System zu klonen,
- das aktuell laufende Live-System auf einen weiteren USB-Datenträger zu übertragen.

Damit ist das Werkzeug insbesondere für die **Bereitstellung eines weiteren bootfähigen Rescue-Mediums** relevant.

Grundlegender Ablauf

Für die Erstellung eines Live-USB-Mediums werden grundsätzlich:

1. Ziel-USB-Datenträger auswählen

2. ISO beziehungsweise andere Quelle auswählen

3. Einstellungen kontrollieren

4. Schreibvorgang starten

benötigt.

Der wichtigste Schritt ist dabei die eindeutige Identifikation des **Zieldatenträgers**.

Full-featured Mode

Für Systeme aus der MX-/antiX-Familie steht ein erweiterter Modus zur Verfügung.

Dieser unterstützt die besonderen Funktionen der MX-/antiX-Live-Infrastruktur, darunter auch Funktionen, die beschreibbare Live-Medien und Persistence verwenden.

Diese MX-spezifischen Funktionen sind für die allgemeine Betriebssystemrettung mit JUMBO BLOCK Rescue nicht weiter zu vertiefen.

Image Mode

Für ISO-Dateien außerhalb der MX-/antiX-Familie verwendet MX Live USB Maker den **Image Mode**.

Dabei wird die ISO mit einem dd-basierten Verfahren als Abbild auf das Zielmedium geschrieben.

MX Live USB Maker führt dabei zusätzliche Plausibilitätsprüfungen durch, um die Gefahr einer Auswahl des falschen Zielmediums zu reduzieren.

Dadurch kann das Werkzeug grundsätzlich auch zum Schreiben geeigneter Live- oder Installations-ISOs anderer Betriebssysteme beziehungsweise Distributionen verwendet werden.

Live-System klonen

MX Live USB Maker kann:

ein vorhandenes Live-System

oder:

das aktuell laufende Live-System

auf einen USB-Datenträger klonen.

Für JUMBO BLOCK Rescue kann dies beispielsweise zur Erstellung eines weiteren bootfähigen Arbeitsmediums genutzt werden.

Dry Run

Die Option **Dry Run** simuliert den vorgesehenen Vorgang, ohne tatsächlich auf den Datenträger zu schreiben.

Diese Funktion ist besonders hilfreich, um die gewählte Konfiguration zunächst zu kontrollieren.

Ein Dry Run ersetzt jedoch nicht die manuelle Kontrolle des ausgewählten Zielgeräts.

Verschlüsseltes Live-USB

MX Live USB Maker unterstützt außerdem die Erstellung eines vollständig mit **LUKS** verschlüsselten Live-USB-Mediums.

Diese Funktion betrifft den Schutz des Live-Mediums und ist keine Funktion zur Entschlüsselung oder Rettung eines fremden Datenträgers.

Für die allgemeine LUKS-Verwaltung enthält JUMBO BLOCK Rescue mit **Cryptsetup** ein separates Werkzeug.

Erweiterte Optionen

MX Live USB Maker besitzt weitere Optionen beispielsweise für:

- GPT-Partitionierung,
- Aktualisierung bestehender Live-USB-Medien,
- Größenbegrenzung des vom Live-System verwendeten Bereichs,
- Dateisystemkennzeichnung,
- spezielle Boot- und Entwicklungsoptionen.

Viele dieser Einstellungen dienen speziellen MX-/antiX- oder Entwicklungsanforderungen.

Sie gehören nicht zum normalen Rescue-Arbeitsablauf und werden deshalb im JUMBO BLOCK Rescue Handbuch nicht weiter vertieft.

Typischer Einsatz

Ein weiteres bootfähiges Rescue-Medium soll aus einer vorhandenen ISO erzeugt werden.

Vorgehen:

1. Geeigneten USB-Datenträger anschließen.

2. MX Live USB Maker starten.

3. Zielgerät anhand von Gerät, Kapazität und Inhalt eindeutig identifizieren.

4. ISO auswählen.

5. Gewählten Modus und Einstellungen kontrollieren.

6. Gegebenenfalls zunächst Dry Run verwenden.

7. Schreibvorgang starten.

8. Vorgang vollständig abschließen lassen.

9. Das erzeugte Medium anschließend durch einen kontrollierten Bootversuch prüfen.

Grenzen

MX Live USB Maker:

- repariert kein installiertes Betriebssystem,
- stellt keine gelöschten Dateien wieder her,
- repariert keine Partitionstabellen,

- diagnostiziert keine Datenträgerdefekte,
- garantiert nicht, dass jede beliebige ISO auf jeder Hardware bootfähig ist.

Ob ein erzeugtes Medium tatsächlich startet, hängt unter anderem von der verwendeten ISO, deren Bootunterstützung und der Zielhardware ab.

Sicherheitshinweise

Das Schreiben eines Images auf einen USB-Datenträger überschreibt vorhandene Daten beziehungsweise Datenträgerstrukturen des ausgewählten Zielmediums.

MX weist auch bei der zugrunde liegenden dd-Methode ausdrücklich darauf hin, dass ein falsch ausgewähltes Zielmedium vollständig überschrieben werden kann.

Deshalb gilt:

Zieldatenträger vor dem Start immer eindeutig anhand von Gerät und Kapazität identifizieren.

Ein Datenträger mit noch benötigten Daten darf nicht als Ziel verwendet werden.

Automatische Sicherheitsprüfungen des Programms ersetzen diese Kontrolle nicht.

Technische Einordnung in JUMBO BLOCK Rescue

MX Live USB Maker gehört zu:

Block B – MX-Werkzeuge → Rescue-Medium

Das Werkzeug repariert kein Zielsystem. Seine Aufgabe besteht vielmehr darin, **bootfähige Live- und Rescue-Medien bereitzustellen oder zu vervielfältigen**.

Besonders relevant ist dabei die Trennung:

Full-featured Mode

→ erweiterte MX-/antiX-Live-Funktionen.

Image Mode

→ ISO-Abbild auf USB schreiben; auch für Nicht-MX-/Nicht-antiX-ISOs vorgesehen.

Der zentrale Sicherheitsgrundsatz lautet:

Quelle und Ziel eindeutig unterscheiden – Zielmedium vor dem Schreiben kontrollieren – vorhandene Daten auf dem Ziel als gefährdet betrachten – erzeugtes Rescue-Medium anschließend durch einen Bootversuch prüfen.

B.5 MX Snapshot

Bereich: Systemabbild / ISO-Erstellung
Programm: MX Snapshot
Baseline 20261001: 26.08
Bedienung: Grafische Benutzeroberfläche
Aufgabe: Einen vorbereiteten MX-basierten Systemstand einschließlich bewusst einbezogener Dateien als bootfähiges ISO-Abbild sichern

Zweck

MX Snapshot erzeugt aus einem laufenden installierten MX-basierten System ein bootfähiges ISO-Abbild.

Innerhalb von JUMBO BLOCK Rescue kann diese Funktion über die reine Reproduktion des Rescue-Systems hinausgehen:

Ausgewählte Dateien eines untersuchten oder beschädigten Systems können zunächst mit den Rescue-Werkzeugen gesichert, kontrolliert in den Snapshot-Bestand übernommen und anschließend gemeinsam mit dem Rescue-System in einem bootfähigen ISO-Abbild gesichert werden.

Damit kann MX Snapshot auch Bestandteil eines Rescue-Arbeitsablaufs zur **geordneten Überführung ausgewählter geretteter Daten** sein.

Einsatzgebiet im Rescue-System

Ein möglicher Anwendungsfall ist ein kompromittiertes oder gehacktes System.

JUMBO BLOCK Rescue kann unabhängig vom betroffenen Betriebssystem verwendet werden, um benötigte Dateien zu identifizieren und auf einen kontrollierten Arbeitsbereich zu übertragen.

Ausgewählte gesicherte Dateien können anschließend bewusst Bestandteil eines mit MX Snapshot erzeugten ISO-Abbilds werden.

Dadurch können in einem definierten Abbild beispielsweise gemeinsam gesichert werden:

- das bootfähige Rescue-System,
- ausgewählte gerettete Dokumente,
- Protokolle,
- Konfigurationsdateien,
- Untersuchungsergebnisse,
- technische Dokumentation,
- weitere bewusst übernommene Dateien.

Welche Dateien tatsächlich einbezogen werden, muss im jeweiligen Rescue-Fall entschieden werden.

Besondere Bedeutung bei kompromittierten Systemen

Dateien aus einem gehackten oder anderweitig kompromittierten System sind grundsätzlich als **potenziell nicht vertrauenswürdig** zu behandeln.

Die Aufnahme in ein Snapshot-ISO macht diese Dateien nicht sicher und verändert ihren Vertrauensstatus nicht.

Insbesondere ausführbare Dateien, Skripte, Makros, manipulierte Dokumente oder andere aktive Inhalte können weiterhin schädlichen Code enthalten.

Deshalb gilt:

Sichern ist nicht gleich Vertrauen.

Die übernommenen Dateien sollten innerhalb des Rescue-Arbeitsablaufs klar vom vertrauenswürdigen Systembestand getrennt und als Material aus dem untersuchten Zielsystem behandelt werden.

System-Snapshot

Für einen weiterzugebenden Systemstand bietet MX Snapshot einen System-Snapshot-Modus.

Dieser übernimmt den installierten Systemstand und die vorgesehenen Anwendungen, ohne einfach sämtliche persönlichen Inhalte eines Benutzerkontos in das resultierende System zu übernehmen.

Welche zusätzlichen Rescue-Dateien Bestandteil des Snapshots werden sollen, muss deshalb bewusst vorbereitet werden.

Personal Snapshot

Der Personal-Snapshot-Modus kann wesentlich mehr Benutzerinhalte berücksichtigen.

Für einen kontrollierten Rescue-Arbeitsablauf ist jedoch nicht die größtmögliche Übernahme von Daten das Ziel.

Entscheidend ist vielmehr:

Nur die tatsächlich vorgesehenen Dateien sollen Bestandteil des erzeugten Abbilds werden.

Ausschlüsse

MX Snapshot verwendet Ausschlussregeln, mit denen Dateien und Verzeichnisse vom erzeugten ISO ferngehalten werden können.

Diese Ausschlüsse sind bei einem Rescue-Snapshot besonders wichtig.

Vor der ISO-Erstellung muss kontrolliert werden:

- welche geretteten Dateien aufgenommen werden sollen,
- welche Arbeitsdateien nicht benötigt werden,

- ob persönliche oder vertrauliche Daten enthalten sind,
- ob temporäre Dateien ausgeschlossen werden sollen,
- ob potenziell gefährliche Dateien bewusst als Untersuchungsmaterial mitgeführt werden.

Typischer Rescue-Ablauf

Ein kompromittiertes System soll nicht weiter produktiv verwendet werden, bestimmte Dateien müssen jedoch gesichert werden.

1. JUMBO BLOCK Rescue unabhängig vom betroffenen System starten.

2. Zustand der Datenträger prüfen.

3. Benötigte Dateien identifizieren und in einen kontrollierten Arbeitsbereich sichern.

4. Bei Bedarf Dateien mit den vorhandenen Analyse- und Sicherheitswerkzeugen untersuchen.

5. Herkunft und Vertrauensstatus der übernommenen Dateien beibehalten beziehungsweise dokumentieren.

6. Nur die tatsächlich benötigten Dateien für den Snapshot vorsehen.

7. Ausschlussregeln kontrollieren.

8. MX Snapshot starten und ISO erzeugen.

9. Erzeugtes ISO auf Integrität und Inhalt kontrollieren.

10. Bootfähigkeit des ISO prüfen.

Damit kann ein transportierbarer Systemstand entstehen, der JUMBO BLOCK Rescue und ausgewähltes Sicherungs- beziehungsweise Untersuchungsmaterial gemeinsam enthält.

Keine automatische Dekontamination

MX Snapshot führt keine Sicherheitsbereinigung der aufgenommenen Dateien durch.

Eine Datei aus einem kompromittierten System bleibt auch innerhalb eines Snapshot-ISO potenziell kompromittiert.

Deshalb sollten solche Dateien nicht allein deshalb geöffnet oder ausgeführt werden, weil sie erfolgreich in ein neues ISO übernommen wurden.

Für eine Untersuchung stehen innerhalb von JUMBO BLOCK Rescue unter anderem die entsprechenden Sicherheits- und Analysewerkzeuge aus Block A zur Verfügung.

Weitere Verwendung

MX Snapshot kann daneben selbstverständlich auch zur:

- Sicherung eines definierten JUMBO BLOCK Rescue Systemstands,
- Reproduktion eines vorbereiteten Systems,

- Erstellung eines angepassten bootfähigen ISO,
- Weitergabe eines definierten Systemstands

eingesetzt werden.

Diese Möglichkeiten stehen neben dem Rescue-Anwendungsfall der kontrollierten Überführung ausgewählter Dateien.

Grenzen

MX Snapshot selbst:

- rettet keine gelöschten Dateien,
- repariert keine beschädigten Dateisysteme,
- untersucht keine Schadsoftware,
- entscheidet nicht über die Vertrauenswürdigkeit übernommener Dateien,
- bereinigt keine kompromittierten Dateien.

Die eigentliche Rettung und Analyse erfolgt mit den dafür vorgesehenen Werkzeugen von JUMBO BLOCK Rescue.

MX Snapshot kann anschließend dazu dienen, **den vorbereiteten System- und Datenbestand in einem bootfähigen ISO-Abbild zusammenzuführen**.

Sicherheitshinweise

Bei Daten aus einem kompromittierten System gilt:

Herkunft erhalten – Vertrauensstatus nicht verändern – Dateien nicht ungeprüft ausführen.

Nur benötigte Dateien übernehmen.

Ausführbare Inhalte besonders kritisch behandeln.

Vor der Snapshot-Erstellung kontrollieren, welche Daten tatsächlich Bestandteil des ISO werden.

Persönliche, vertrauliche und nicht benötigte Daten ausschließen.

Das fertige ISO auf Inhalt, Integrität und Bootfähigkeit prüfen.

Technische Einordnung in JUMBO BLOCK Rescue

MX Snapshot gehört zu:

Block B – MX-Werkzeuge → Systemabbild / ISO-Erstellung

Seine Rescue-Bedeutung kann über die reine Reproduktion des Basissystems hinausgehen.

Es ermöglicht nach entsprechender Vorbereitung, einen **definierten bootfähigen Rescue-Systemstand zusammen mit bewusst ausgewählten geretteten oder zu untersuchenden Dateien in einem ISO-Abbild zu sichern und zu überführen**.

Damit ergibt sich eine klare Aufgabenteilung:

JUMBO BLOCK Rescue Werkzeuge

→ identifizieren, diagnostizieren, retten, kopieren und untersuchen Daten.

MX Snapshot

→ kann den anschließend vorbereiteten System- und Datenbestand in einem bootfähigen ISO-Abbild zusammenführen.

Der zentrale Grundsatz lautet:

Gerettete Daten kontrolliert übernehmen – Herkunft und Vertrauensstatus bewahren – nur benötigte Dateien einbinden – Snapshot nicht mit Dekontamination verwechseln – fertiges ISO anschließend prüfen.

B.6 ISO Signature Verifier

Bereich: Integritäts- und Authentizitätsprüfung
Programm: ISO Signature Verifier
Baseline 20261001: 2026.09.01
Paket: verify-iso-sig
Bedienung: Grafische Benutzeroberfläche / Kommandozeile
Aufgabe: Signaturen und Prüfsummen von ISO-Abbildern verifizieren

Zweck

ISO Signature Verifier dient zur Überprüfung von ISO-Abbildern anhand der zugehörigen kryptografischen Signaturen und Prüfsummeninformationen.

Damit kann vor der Verwendung eines ISO geprüft werden, ob die vorliegende Datei mit den vom Herausgeber signierten beziehungsweise veröffentlichten Referenzinformationen übereinstimmt.

Das Werkzeug verbindet dabei zwei unterschiedliche Prüfprinzipien:

- kryptografische Überprüfung einer Signatur,
- Integritätsprüfung des ISO-Abbilds anhand einer Prüfsumme.

Bedeutung für JUMBO BLOCK Rescue

ISO-Dateien können innerhalb eines Rescue-Systems eine besondere sicherheitstechnische Bedeutung besitzen.

Sie können beispielsweise:

- Betriebssysteme,
- Installationsmedien,
- Live-Systeme,
- Diagnoseumgebungen,
- Rescue-Systeme,
- Firmware- oder Wartungsumgebungen

enthalten.

Bevor ein solches ISO gestartet, installiert, auf einen Datenträger übertragen oder anderweitig eingesetzt wird, sollte seine Integrität und – sofern entsprechende Signaturinformationen vorliegen – seine kryptografisch belegbare Herkunft geprüft werden.

ISO Signature Verifier stellt hierfür innerhalb von JUMBO BLOCK Rescue ein spezialisiertes Prüfwerkzeug bereit.

Betriebssystemunabhängige Prüfung

ISO Signature Verifier prüft nicht, welches Betriebssystem sich in einem ISO-Abbild befindet.

Gegenstand der Prüfung sind die zum ISO gehörenden **Signatur- und Prüfsummeninformationen**.

Entscheidend ist daher nicht die Betriebssystemfamilie, sondern ob für das betreffende ISO geeignete und vom Werkzeug unterstützte Signatur- beziehungsweise Prüfsummeninformationen zur Verfügung stehen und der verwendete Signaturschlüssel zuverlässig dem erwarteten Herausgeber zugeordnet werden kann.

Damit ist das zugrunde liegende Prüfprinzip **betriebssystemunabhängig**.

Für JUMBO BLOCK Rescue gilt:

Nicht das Betriebssystem im ISO entscheidet über dessen Verifizierbarkeit, sondern das verwendete Signatur- und Prüfsummenverfahren sowie die Verfügbarkeit einer vertrauenswürdigen Referenz.

Prüfsumme und Signatur

Prüfsumme und kryptografische Signatur erfüllen unterschiedliche Aufgaben.

Eine **Prüfsumme** ermöglicht den Vergleich einer Datei mit einem bekannten Referenzwert.

Stimmt beispielsweise eine veröffentlichte SHA-Prüfsumme mit der aus dem vorliegenden ISO berechneten Prüfsumme überein, entspricht die geprüfte Datei dem Datenbestand, für den dieser Referenzwert erzeugt wurde.

Eine Prüfsumme allein beantwortet jedoch nicht automatisch die Frage, **wer diesen Referenzwert veröffentlicht hat**.

Hier kommt die kryptografische Signatur hinzu.

Eine verifizierbare Signatur kann die Prüfsumme beziehungsweise das ISO kryptografisch mit einem bestimmten Signaturschlüssel verbinden.

Damit werden:

Integrität

und:

Authentizitätsprüfung

miteinander kombiniert.

Unterschiedliche Signaturverfahren

Herausgeber verwenden für ISO-Abbilder unterschiedliche Veröffentlichungs- und Signaturverfahren.

Je nach Verfahren kann beispielsweise:

- das ISO unmittelbar signiert sein,

- eine Prüfsummendatei signiert sein,
- eine einzelne zum ISO gehörende Prüfsumme signiert sein,
- Prüfsumme und Signaturinformation gemeinsam veröffentlicht werden.

ISO Signature Verifier wertet die unterstützten Strukturen aus und führt die dazugehörigen Prüfungen durch.

Grundlegender Arbeitsablauf

Die Bedienung folgt grundsätzlich einem einfachen Schema:

1. ISO Signature Verifier starten.

2. Zu prüfendes ISO beziehungsweise die zugehörige Signaturinformation auswählen.

3. Verifikation starten.

4. Prüfergebnis kontrollieren.

Sind die erforderlichen Begleitdateien vorhanden und wird das verwendete Verfahren unterstützt, können ISO, Prüfsumme und Signatur miteinander abgeglichen werden.

Signaturschlüssel

Eine kryptografisch korrekte Signatur ist nur dann aussagekräftig, wenn auch geklärt ist, **wem der verwendete Signaturschlüssel tatsächlich gehört**.

Bei einem unbekannten Schlüssel muss deshalb dessen Identität kontrolliert werden.

Dafür sind insbesondere:

- Key ID,
- Fingerprint,
- angegebene Identität

relevant.

Der Fingerprint eines unbekannten Schlüssels sollte über eine unabhängige und vertrauenswürdige Quelle des jeweiligen Herausgebers überprüft werden.

Gültige Signatur bedeutet nicht automatisch vertrauenswürdiger Herausgeber

Eine wichtige Unterscheidung lautet:

Kryptografisch gültige Signatur

bedeutet zunächst, dass die Signatur technisch zu den geprüften Daten und dem verwendeten Schlüssel passt.

Ob dieser Schlüssel tatsächlich zum erwarteten Herausgeber gehört, ist eine weitere Frage.

Deshalb dürfen unbekannte Signaturschlüssel nicht allein aufgrund einer technisch erfolgreichen Signaturprüfung als vertrauenswürdig betrachtet werden.

Typischer Rescue-Ablauf

Ein externes ISO soll innerhalb von JUMBO BLOCK Rescue verwendet oder auf ein bootfähiges Medium übertragen werden.

1. ISO aus einer nachvollziehbaren Quelle beziehen.

2. Vorhandene Signatur- und Prüfsummeninformationen ebenfalls beschaffen.

3. ISO Signature Verifier starten.

4. ISO und zugehörige Verifikationsinformationen prüfen.

5. Ergebnis kontrollieren.

6. Bei einem unbekannten Signaturschlüssel dessen Fingerprint unabhängig überprüfen.

7. ISO erst nach erfolgreicher Prüfung weiterverwenden.

8. Bei Bedarf das geprüfte ISO anschließend mit MX Live USB Maker auf ein geeignetes USB-Medium übertragen.

Damit ergibt sich ein klarer Arbeitsablauf:

ISO beschaffen → Herkunft und Integrität verifizieren → erst danach einsetzen.

Fehlgeschlagene Verifikation

Schlägt eine Verifikation fehl, kann dies unterschiedliche Ursachen haben.

Möglich sind beispielsweise:

- beschädigte Datei,
- unvollständige Datei,
- verändertes ISO,
- falsche Prüfsummendatei,
- falsche Signaturdatei,
- nicht zusammengehörende Dateien,
- nicht passender Signaturschlüssel.

Aus dem Fehlschlagen der Verifikation allein lässt sich deshalb nicht automatisch ableiten, **warum** die Prüfung fehlgeschlagen ist.

Das betreffende ISO sollte jedoch nicht ungeprüft weiterverwendet werden.

Erfolgreiche Verifikation

Eine erfolgreiche Verifikation bestätigt im Rahmen des verwendeten Prüfverfahrens, dass das ISO mit den geprüften Signatur- beziehungsweise Prüfsummeninformationen übereinstimmt.

Für eine belastbare Aussage zur Herkunft ist zusätzlich erforderlich, dass der verwendete Signaturschlüssel zuverlässig dem erwarteten Herausgeber zugeordnet werden kann.

Keine Malware-Prüfung

ISO Signature Verifier ist kein Viren- oder Malware-Scanner.

Eine erfolgreiche Verifikation bedeutet nicht automatisch, dass sämtliche Inhalte eines ISO sicher oder frei von Schadsoftware sind.

Sie bedeutet insbesondere nicht, dass:

- keine Sicherheitslücken enthalten sind,
- sämtliche enthaltene Software vertrauenswürdig ist,
- das ISO auf der vorhandenen Hardware fehlerfrei funktioniert,
- das ISO tatsächlich bootfähig ist.

Das Werkzeug beantwortet eine andere Frage:

Stimmt das vorliegende ISO mit den überprüften Signatur- und Prüfsummeninformationen überein?

Zusammenspiel mit ClamAV

ISO Signature Verifier und ClamAV erfüllen unterschiedliche Sicherheitsaufgaben.

ISO Signature Verifier

→ prüft Integrität und kryptografische Authentizität anhand vorhandener Referenzinformationen.

ClamAV

→ untersucht zugängliche Dateien auf erkennbare Schadsoftware.

Beide Prüfungen dürfen deshalb nicht miteinander verwechselt werden.

Zusammenspiel mit MX Live USB Maker

Auch ISO Signature Verifier und MX Live USB Maker haben klar getrennte Aufgaben.

ISO Signature Verifier

→ überprüft ein ISO.

MX Live USB Maker

→ überträgt ein ISO auf ein USB-Medium.

Ein sinnvoller Ablauf innerhalb von JUMBO BLOCK Rescue lautet daher:

ISO verifizieren → anschließend auf das Zielmedium übertragen → erzeugtes Medium testen.

Eigene Prüfsummen

Für eine vorhandene Datei kann jederzeit eine neue Prüfsumme berechnet werden.

Eine solche Prüfsumme ist nützlich, um später festzustellen, ob sich genau diese Datei verändert hat.

Sie beweist jedoch nicht rückwirkend die Herkunft der ursprünglichen Datei.

Dafür ist eine vertrauenswürdige Referenz erforderlich.

Grenzen

ISO Signature Verifier kann eine Authentizitätsprüfung nur durchführen, wenn geeignete Signaturinformationen zur Verfügung stehen.

Sind keine Signaturinformationen vorhanden, kann eine vorhandene veröffentlichte Prüfsumme gegebenenfalls noch zur Integritätskontrolle dienen.

Ohne eine vertrauenswürdige Referenz lässt sich jedoch allein aus einer selbst berechneten Prüfsumme nicht bestimmen, von welchem Herausgeber ein ISO stammt.

Das Werkzeug ersetzt außerdem weder eine Sicherheitsanalyse des ISO-Inhalts noch einen praktischen Boot- und Funktionstest.

Sicherheitshinweise

Signatur- und Prüfsummeninformationen möglichst aus einer vertrauenswürdigen Quelle des Herausgebers beziehen.

Unbekannte Signaturschlüssel nicht ungeprüft akzeptieren.

Fingerprints unbekannter Schlüssel unabhängig kontrollieren.

Ein ISO mit fehlgeschlagener Verifikation nicht ohne weitere Klärung einsetzen.

Eine erfolgreiche Signaturprüfung nicht mit einer vollständigen Sicherheitsanalyse verwechseln.

Vor produktiver Verwendung eines bootfähigen ISO zusätzlich dessen tatsächliche Boot- und Funktionsfähigkeit prüfen.

Technische Einordnung in JUMBO BLOCK Rescue

ISO Signature Verifier gehört zu:

Block B – MX-Werkzeuge → Integritäts- und Authentizitätsprüfung

Seine Funktion ist **nicht auf eine bestimmte Betriebssystemfamilie ausgerichtet**.

Entscheidend ist das für das betreffende ISO verwendete Signatur- und Prüfsummenverfahren.

Damit erfüllt ISO Signature Verifier innerhalb von JUMBO BLOCK Rescue eine wichtige Kontrollfunktion:

Ein externes ISO kann vor seiner Verwendung auf Übereinstimmung mit vertrauenswürdigen kryptografischen Referenzinformationen geprüft werden.

In Verbindung mit MX Live USB Maker ergibt sich daraus ein sicherer Grundablauf:

ISO beschaffen → Signatur und Prüfsumme verifizieren → ISO auf Rescue-Medium übertragen → Medium praktisch testen.

Der zentrale Grundsatz lautet:

Nicht dem Dateinamen oder der vermeintlichen Herkunft eines ISO vertrauen – verfügbare kryptografische Referenzen prüfen – unbekannte Schlüssel verifizieren – erst danach das Abbild einsetzen.

Block C

Neben den spezialisierten Rescue-Werkzeugen aus **Block A – JB-Rescue** und den ausgewählten MX-Werkzeugen aus **Block B** stellt JUMBO BLOCK Rescue verschiedene allgemeine Arbeitswerkzeuge bereit.

Diese Programme übernehmen keine zentrale Rescue-Funktion, unterstützen jedoch die praktische Arbeit während eines Einsatzes. Dazu gehören insbesondere das Formatieren von Wechselmedien, die grafische Arbeit mit Dateien und Archiven, die Suche nach Dateien, die Bearbeitung von Textdateien, die Prüfung von Prüfsummen, die Netzwerkkonfiguration sowie Terminal- und Dokumentationsaufgaben.

Für dieses Handbuch werden nur diejenigen Zubehörprogramme einzeln beschrieben, die einen konkreten praktischen Nutzen innerhalb einer Rescue-Umgebung besitzen. Weitere Programme und Desktop-Komponenten des Basissystems sind nicht Gegenstand der ausführlichen Dokumentation.

Werkzeuggruppen

Neun relevante Zubehör-Werkzeuge:

C.1 – Format USB
Formatierung und Vorbereitung von USB-Sticks und anderen geeigneten Wechselmedien für die weitere Verwendung.

C.2 – Thunar File Manager
Grafische Verwaltung von Dateien, Verzeichnissen, Datenträgern und eingebundenen Dateisystemen.

C.3 – Catfish File Search
Gezielte Suche nach Dateien innerhalb zugänglicher Verzeichnisse und eingebundener Datenträger.

C.4 – Engrampa Archive Manager
Grafische Arbeit mit unterstützten Archivformaten zum Öffnen, Erstellen und Entpacken von Archiven.

C.5 – FeatherPad
Leichtgewichtiger Texteditor zur Anzeige und Bearbeitung von Text-, Konfigurations- und Protokolldateien.

C.6 – GtkHash
Berechnung und Vergleich kryptografischer Prüfsummen zur Kontrolle der Integrität von Dateien.

C.7 – Advanced Network Configuration
Grafische Konfiguration und Bearbeitung von Netzwerkverbindungen für die Kommunikation innerhalb einer Rescue-Umgebung.

C.8 – Xfce Terminal
Terminalumgebung für Kommandozeilenwerkzeuge, Diagnosebefehle und administrative Arbeiten.

C.9 – Screenshot
Erstellung von Bildschirmaufnahmen zur Dokumentation von Systemzuständen, Fehlermeldungen, Analyseergebnissen und durchgeführten Rescue-Arbeiten.

Die neun Zubehör-Werkzeuge ergänzen damit die spezialisierten Funktionen aus **Block A – JB-Rescue** und **Block B – MX-Werkzeuge**, ohne selbst den Kern des Rescue-Systems zu bilden.

Block C stellt die allgemeinen Arbeitsmittel bereit, die Untersuchung, Sicherung, Bearbeitung und Dokumentation während eines Rescue-Einsatzes praktisch unterstützen.

Zuordnung der Bereiche, Programme und Versionen

Nr.	Bereich	Programm	Baseline-Version
C.1	Datenträger	Format USB	26.09.01 ff.
C.2	Dateien	Thunar File Manager	4.20.2 ff.
C.3	Dateien	Catfish File Search	4.20.1 ff.
C.4	Archive	Engrampa Archive Manager	1.26.2 ff.
C.5	Text	FeatherPad	1.6.2 ff.
C.6	Integrität	GtkHash	1.5 ff.
C.7	Netzwerk	Advanced Network Configuration	1.36.0 ff.
C.8	System	Xfce Terminal	1.1.4 ff.
C.9	Dokumentation	Screenshot	1.11.1 ff.

C.1 – Format USB

Bereich: Datenträger
Programm: Format USB
Baseline 20261001: 26.09.01 ff.

Zweck

Format USB ist ein grafisches Werkzeug zum Formatieren von USB-Wechselmedien.

Das Programm ermöglicht die Auswahl eines angeschlossenen USB-Datenträgers, eines Dateisystems und optional einer Datenträgerbezeichnung. Dabei wird das ausgewählte Gerät neu partitioniert und mit einem neuen Dateisystem versehen.

Innerhalb von JUMBO BLOCK Rescue dient Format USB insbesondere dazu, USB-Medien für die Aufnahme, Übertragung oder Weitergabe von Dateien vorzubereiten.

Einsatzgebiet im Rescue-System

Während eines Rescue-Einsatzes werden häufig zusätzliche Wechselmedien benötigt, beispielsweise für:

- gerettete Dateien,
- Protokolle und Untersuchungsergebnisse,
- exportierte Konfigurationen,
- Screenshots und Dokumentationen,
- Arbeitskopien,
- Dateien für die Übergabe an ein anderes System.

Format USB stellt hierfür ein einfaches grafisches Werkzeug zur Vorbereitung eines geeigneten USB-Datenträgers bereit.

Es ist dabei von **MX Live USB Maker aus Block B** zu unterscheiden:

Format USB formatiert einen USB-Datenträger für die allgemeine Verwendung.

MX Live USB Maker erzeugt dagegen ein bootfähiges Medium aus einem ISO-Abbild beziehungsweise einem unterstützten Live-System.

Funktionen

Nach der offiziellen MX-Dokumentation unterstützt Format USB folgende Dateisysteme:

- **FAT32/VFAT**
- **NTFS**
- **ext4**

- **exFAT**

Zusätzlich kann eine Bezeichnung für das neu erzeugte Dateisystem vergeben werden.

Das ausgewählte USB-Gerät wird dabei als Ganzes neu partitioniert. Anschließend wird eine Partition angelegt und mit dem gewählten Dateisystem formatiert.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue besteht aus folgenden Schritten:

1. Gewünschtes USB-Medium anschließen.
2. **Format USB** starten.
3. Das richtige Zielgerät eindeutig identifizieren.
4. Gewünschtes Dateisystem auswählen.
5. Bei Bedarf eine Datenträgerbezeichnung vergeben.
6. Auswahl nochmals kontrollieren.
7. Formatierung ausführen.
8. Nach Abschluss prüfen, ob das Medium korrekt eingebunden und verwendet werden kann.
9. Erst danach die zu sichernden oder zu übertragenden Dateien auf das Medium kopieren.

Besonders wichtig ist die eindeutige Identifikation des Zielgeräts. Bei mehreren angeschlossenen USB-Datenträgern sollte die Zuordnung anhand der angezeigten Geräteinformationen sorgfältig kontrolliert werden.

Wahl des Dateisystems

Welches Dateisystem verwendet wird, hängt vom vorgesehenen Zielsystem und Einsatzzweck ab.

FAT32/VFAT bietet eine sehr breite Kompatibilität, besitzt jedoch technische Einschränkungen, unter anderem hinsichtlich der Größe einzelner Dateien.

exFAT eignet sich insbesondere für den Austausch größerer Dateien zwischen unterschiedlichen modernen Betriebssystemen.

NTFS kann sinnvoll sein, wenn das Medium überwiegend mit Systemen verwendet werden soll, die dieses Dateisystem unterstützen.

ext4 ist ein Linux-Dateisystem und eignet sich insbesondere für eine weitere Verwendung innerhalb entsprechender Linux-Umgebungen.

Für Rescue-Arbeiten sollte das Dateisystem daher nicht allein nach verfügbarer Speicherkapazität, sondern auch nach dem vorgesehenen Zielsystem gewählt werden.

Typische Rescue-Szenarien

Bereitstellung eines Datenträgers für gerettete Dateien

Nach einer Datenrettung mit TestDisk, PhotoRec, Foremost oder anderen Werkzeugen kann ein separates USB-Medium vorbereitet werden, auf das die geretteten Daten anschließend übertragen werden.

Übergabe von Untersuchungsergebnissen

Protokolle, Prüfsummen, Screenshots, Konfigurationsdateien oder andere Untersuchungsergebnisse können auf einem neu vorbereiteten USB-Datenträger gesammelt und anschließend auf einem anderen System ausgewertet werden.

Datenaustausch zwischen unterschiedlichen Betriebssystemen

Durch die Auswahl eines geeigneten Dateisystems kann ein Wechselmedium für den Datenaustausch zwischen JUMBO BLOCK Rescue und einem anderen Betriebssystem vorbereitet werden.

Grenzen und Sicherheitshinweise

Format USB ist **kein Datenrettungswerkzeug**.

Das Programm stellt keine verlorenen Dateien wieder her, repariert keine beschädigten Dateisysteme und erstellt keine sektorweisen Sicherungsabbilder.

Besonders wichtig:

Die Formatierung verändert das ausgewählte Zielmedium grundlegend. Vorhandene Partitionen und darauf gespeicherte Daten sind dadurch gefährdet beziehungsweise werden entfernt.

Ein Datenträger, von dem noch Daten gerettet werden sollen, darf deshalb nicht mit Format USB vorbereitet werden.

Ebenso darf ein beschädigter oder instabiler Quelldatenträger nicht versehentlich als Ziel ausgewählt werden.

Bestehen Zweifel an der Gerätezuordnung, sollte die Identität des Datenträgers zunächst mit den Diagnosewerkzeugen aus Block A überprüft werden.

Bei einem möglicherweise kompromittierten System gilt außerdem:

Das Formatieren eines USB-Mediums macht darauf anschließend kopierte Dateien nicht automatisch vertrauenswürdig.

Gerettete Dateien aus einem kompromittierten System können weiterhin Schadcode, manipulierte Inhalte oder andere sicherheitsrelevante Veränderungen enthalten.

Technische Einordnung in JUMBO BLOCK Rescue

Format USB gehört nicht zum eigentlichen Datenrettungskern von JUMBO BLOCK Rescue. Es stellt vielmehr ein praktisches Hilfsmittel zur Vorbereitung zusätzlicher Wechselmedien dar.

Die funktionale Trennung ist eindeutig:

Block A – JB-Rescue
identifiziert, untersucht, sichert und rettet Daten und Datenträger.

Block B – MX Live USB Maker
erstellt bootfähige USB-Medien.

C.1 – Format USB
bereitet allgemeine USB-Wechselmedien durch Neupartitionierung und Formatierung für die weitere Verwendung vor.

Damit unterstützt Format USB insbesondere die kontrollierte Aufnahme und Überführung von Daten nach einer Rescue- oder Analysemaßnahme.

Zentrale Regel:

Zielmedium eindeutig identifizieren – vorhandene Daten als gefährdet betrachten – geeignetes Dateisystem auswählen – erst nach erfolgreicher Formatierung die zu sichernden Daten übertragen.

C.2 – Thunar File Manager

Bereich: Dateien
Programm: Thunar File Manager
Baseline 20261001: 4.20.2 ff.

Zweck

Thunar ist der grafische Dateimanager der Xfce-Arbeitsumgebung. Er dient zum Anzeigen, Navigieren, Kopieren, Verschieben, Umbenennen und Verwalten von Dateien und Verzeichnissen.

Innerhalb von JUMBO BLOCK Rescue stellt Thunar die grafische Arbeitsoberfläche für den Umgang mit zugänglichen Dateisystemen und Wechselmedien bereit.

Einsatzgebiet im Rescue-System

Nach der Identifikation und gegebenenfalls Sicherung eines Datenträgers müssen Dateien häufig gesichtet, geordnet oder auf ein anderes Medium übertragen werden.

Thunar eignet sich insbesondere für:

- Navigation durch zugängliche Verzeichnisstrukturen,
- Sichtung geretteter Dateien,
- Kopieren von Dateien auf Sicherungsmedien,
- Übertragung zwischen eingebundenen Datenträgern,
- Anlegen von Arbeits- und Übergabeverzeichnissen,
- Umbenennen und Ordnen geretteter Dateien,
- Öffnen von Dateien mit geeigneten Anwendungen,
- Zugriff auf eingebundene Wechselmedien und Dateisysteme.

Thunar ergänzt damit die spezialisierten Datenrettungswerkzeuge aus Block A um eine grafische Dateiverwaltung.

Funktionen

Zu den grundlegenden Funktionen gehören:

- Anzeige von Dateien und Verzeichnissen,
- Navigation innerhalb von Verzeichnisstrukturen,
- Kopieren und Verschieben,
- Umbenennen,
- Löschen,

- Anlegen neuer Verzeichnisse,
- Anzeige von Dateieigenschaften,
- Öffnen von Dateien mit zugeordneten Anwendungen,
- Arbeit mit eingebundenen Datenträgern und Wechselmedien.

Die konkrete Verfügbarkeit einzelner Funktionen hängt von den Zugriffsrechten, dem eingebundenen Dateisystem und den im Rescue-System vorhandenen Erweiterungen ab.

Bedienung und Arbeitsablauf

Ein typischer Rescue-Arbeitsablauf kann folgendermaßen aussehen:

1. Quelldatenträger identifizieren und dessen Zustand beurteilen.
2. Bei instabilen oder beschädigten Datenträgern zunächst ein Image beziehungsweise eine Sicherung erstellen.
3. Geeigneten Datenträger oder die Arbeitskopie einbinden.
4. Thunar öffnen.
5. Gewünschtes Quellverzeichnis aufrufen.
6. Zu sichernde Dateien auswählen.
7. Separates Zielmedium beziehungsweise Zielverzeichnis öffnen.
8. Dateien kopieren.
9. Kopiervorgang vollständig abschließen lassen.
10. Übertragene Daten anschließend am Ziel kontrollieren.

Für Rescue-Arbeiten sollte grundsätzlich **kopiert statt verschoben** werden, solange die Quelle noch als Original oder Beweismittel erhalten bleiben soll.

Typische Rescue-Szenarien

Übernahme geretteter Dateien

Mit PhotoRec, Foremost oder anderen Datenrettungswerkzeugen wiederhergestellte Dateien können mit Thunar gesichtet, sortiert und anschließend auf ein separates Sicherungsmedium kopiert werden.

Sicherung zugänglicher Benutzerdaten

Ist ein nicht mehr startfähiges Betriebssystem über JUMBO BLOCK Rescue noch lesbar, können zugängliche Dokumente und andere benötigte Dateien grafisch auf ein separates Medium kopiert werden.

Zusammenstellung von Untersuchungsergebnissen

Protokolle, Screenshots, Konfigurationsdateien, Prüfsummen und andere Ergebnisse einer Untersuchung können in einer übersichtlichen Verzeichnisstruktur gesammelt und anschließend übergeben werden.

Datenaustausch mit Wechselmedien

Ein beispielsweise mit **C.1 – Format USB** vorbereitetes Wechselmedium kann in Thunar geöffnet und zur Aufnahme der zu übertragenden Daten verwendet werden.

Grenzen und Sicherheitshinweise

Thunar ist **kein Datenrettungsprogramm**.

Der Dateimanager rekonstruiert keine verlorenen Partitionen, stellt keine gelöschten Dateien wieder her und erzeugt keine sektorweisen Abbilder beschädigter Datenträger.

Kann Thunar eine Datei oder ein Verzeichnis nicht anzeigen, bedeutet dies deshalb nicht automatisch, dass die betreffenden Daten nicht mehr vorhanden oder nicht rettbar sind.

Bei einem instabilen Datenträger sollte nicht versucht werden, große Datenbestände wiederholt direkt mit dem Dateimanager auszulesen. In diesem Fall ist zunächst ein geeignetes Sicherungsabbild, beispielsweise mit GNU ddrescue, vorzuziehen.

Besondere Vorsicht gilt bei schreibenden Operationen:

Verschieben, Umbenennen, Löschen und das Anlegen oder Verändern von Dateien können den Zustand des eingebundenen Dateisystems verändern.

Soll eine Quelle unverändert erhalten bleiben, sollte sie daher möglichst nur lesend eingebunden und ausschließlich auf eine separate Arbeits- oder Sicherungskopie geschrieben werden.

Dateien aus einem kompromittierten System bleiben außerdem potenziell nicht vertrauenswürdig. Das Kopieren mit Thunar verändert ihren Sicherheitsstatus nicht.

Technische Einordnung in JUMBO BLOCK Rescue

Thunar bildet die grafische Dateiverwaltung innerhalb der Rescue-Arbeitsumgebung.

Die funktionale Trennung zu den Werkzeugen aus Block A ist dabei wichtig:

TestDisk, PhotoRec, GNU ddrescue, Foremost und Extundelete
dienen der Sicherung beziehungsweise Wiederherstellung von Daten und Datenträgerinhalten.

Midnight Commander
stellt eine terminalbasierte Dateiverwaltung bereit.

C.2 – Thunar File Manager
ermöglicht die grafische Sichtung, Organisation und Übertragung bereits zugänglicher Dateien und Verzeichnisse.

Damit ist Thunar insbesondere nach einer erfolgreichen Sicherung, Einbindung oder Datenrettung ein praktisches Arbeitswerkzeug für die weitere Behandlung der gewonnenen Daten.

Zentrale Regel:

Original möglichst unverändert erhalten – bei gefährdeten Datenträgern zuerst sichern – gerettete Daten bevorzugt kopieren statt verschieben – Ziel und Quelle vor jeder Dateioperation eindeutig kontrollieren.

C.3 – Catfish File Search

Bereich: Dateien
Programm: Catfish File Search
Baseline 20261001: 4.20.1 ff.

Zweck

Catfish File Search ist ein grafisches Werkzeug zur Suche nach Dateien und Verzeichnissen.

Innerhalb von JUMBO BLOCK Rescue unterstützt Catfish insbesondere die gezielte Suche in zugänglichen Verzeichnisstrukturen, eingebundenen Datenträgern und bereits geretteten Datenbeständen.

Das Programm ergänzt damit den Dateimanager Thunar um eine komfortable Suchfunktion.

Einsatzgebiet im Rescue-System

Bei Rescue-Arbeiten liegen häufig große oder unübersichtliche Datenbestände vor. Gesuchte Dokumente, Konfigurationsdateien, Protokolle oder andere relevante Dateien müssen dann innerhalb umfangreicher Verzeichnisstrukturen lokalisiert werden.

Catfish kann beispielsweise eingesetzt werden, um:

- bestimmte Dateien anhand ihres Namens zu finden,
- Dateitypen beziehungsweise Dateinamensmuster einzugrenzen,
- bestimmte Verzeichnisbereiche gezielt zu durchsuchen,
- gerettete Datenbestände nach relevanten Dateien zu durchsuchen,
- Dateien eines nicht mehr startfähigen, aber noch lesbaren Systems aufzufinden,
- Untersuchungsergebnisse für die weitere Bearbeitung zusammenzustellen.

Funktionen

Catfish stellt eine grafische Suchoberfläche bereit und ermöglicht insbesondere die Suche nach Datei- und Verzeichnisnamen.

Je nach Suchvorgabe und verfügbarer Systemumgebung können Suchergebnisse zusätzlich anhand verschiedener Kriterien eingegrenzt werden.

Der entscheidende Punkt für Rescue-Arbeiten ist:

Catfish durchsucht Dateien und Verzeichnisse, die über das aktuell zugängliche Dateisystem erreichbar sind.

Das Werkzeug arbeitet damit auf einer anderen Ebene als Programme zur Datenrekonstruktion.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue besteht aus folgenden Schritten:

1. Betroffenen Datenträger beziehungsweise eine Arbeitskopie einbinden.
2. Prüfen, ob die benötigte Verzeichnisstruktur zugänglich ist.
3. Catfish starten.
4. Den zu durchsuchenden Verzeichnisbereich festlegen.
5. Suchbegriff beziehungsweise gewünschtes Suchmuster eingeben.
6. Suche durchführen.
7. Trefferliste prüfen.
8. Relevante Dateien bei Bedarf mit Thunar öffnen, kopieren oder in ein separates Arbeitsverzeichnis übernehmen.

Bei großen Datenbeständen empfiehlt es sich, den Suchbereich möglichst sinnvoll einzugrenzen.

Typische Rescue-Szenarien

Suche nach Benutzerdokumenten

Ist ein Betriebssystem nicht mehr startfähig, sein Dateisystem aber weiterhin lesbar, kann Catfish bei der Suche nach benötigten Dokumenten innerhalb der vorhandenen Benutzerverzeichnisse helfen.

Suche in geretteten Datenbeständen

Nach einer Datenrettung können zahlreiche Dateien und Verzeichnisse vorliegen. Catfish unterstützt dabei, innerhalb des zugänglichen Ergebnisses gezielt nach bestimmten Dateinamen oder Dateigruppen zu suchen.

Auffinden von Konfigurations- und Protokolldateien

Bei der Analyse eines Systems können bestimmte Konfigurations-, Protokoll- oder Sicherungsdateien benötigt werden. Sind deren genaue Speicherorte nicht bekannt, kann eine gezielte Suche die weitere Untersuchung erleichtern.

Vorbereitung einer Datenübergabe

Vor einer Übergabe geretteter Daten kann Catfish verwendet werden, um bestimmte gesuchte Dateien innerhalb des Arbeitsbestandes zu lokalisieren. Die eigentliche Übernahme auf ein separates Medium erfolgt anschließend beispielsweise mit Thunar.

Grenzen und Sicherheitshinweise

Catfish ist **kein Datenrettungswerkzeug**.

Eine Datei, die nicht in den Suchergebnissen erscheint, ist deshalb nicht automatisch unwiederbringlich verloren.

Insbesondere kann Catfish keine:

- gelöschten Dateien rekonstruieren,
- verlorenen Partitionen wiederherstellen,
- beschädigten Dateisystemstrukturen reparieren,
- nicht mehr zugänglichen Datenblöcke rekonstruieren,
- sektorweisen Datenträgerabbilder erstellen.

Für solche Aufgaben sind die spezialisierten Werkzeuge aus **Block A – JB-Rescue** vorgesehen.

Auch die Zugriffsrechte des laufenden Systems sowie die tatsächliche Einbindung und Lesbarkeit des untersuchten Dateisystems beeinflussen, welche Daten durchsucht werden können.

Bei einem physisch instabilen oder stark beschädigten Datenträger sollte außerdem nicht mit wiederholten umfangreichen Suchvorgängen auf dem Original gearbeitet werden. In diesem Fall gilt weiterhin das Rescue-Grundprinzip:

Zuerst sichern beziehungsweise ein Image erstellen und anschließend auf der Arbeitskopie suchen.

Technische Einordnung in JUMBO BLOCK Rescue

Catfish übernimmt innerhalb von Block C eine klar abgegrenzte Hilfsfunktion:

Block A – JB-Rescue
rettet beziehungsweise rekonstruiert Daten und macht beschädigte oder verlorene Datenbestände soweit möglich wieder zugänglich.

C.2 – Thunar File Manager
ermöglicht die grafische Navigation, Organisation und Übertragung zugänglicher Dateien.

C.3 – Catfish File Search
unterstützt das gezielte Auffinden von Dateien innerhalb solcher zugänglichen Datenbestände.

Catfish ist damit insbesondere dann sinnvoll, wenn Daten bereits zugänglich oder erfolgreich gerettet wurden, ihr konkreter Speicherort innerhalb eines größeren Bestandes jedoch noch ermittelt werden muss.

Zentrale Regel:

Suchbereich gezielt wählen – Originale möglichst unverändert lassen – bei instabilen Datenträgern zuerst sichern – ein fehlender Suchtreffer ist kein Nachweis dafür, dass eine Datei nicht mehr rettbar ist.

C.4 – Engrampa Archive Manager

Bereich: Archive
Programm: Engrampa Archive Manager
Baseline 20261001: 1.26.2 ff.

Zweck

Engrampa Archive Manager ist ein grafisches Werkzeug zum Erstellen, Öffnen und Bearbeiten von Archivdateien.

Innerhalb von JUMBO BLOCK Rescue dient Engrampa insbesondere dazu, vorhandene oder gerettete Archive zu öffnen, deren Inhalt einzusehen, ausgewählte Dateien zu extrahieren sowie Daten für eine geordnete Übergabe in Archiven zusammenzufassen.

Einsatzgebiet im Rescue-System

Bei Rescue- und Analysearbeiten können Archivdateien sowohl Bestandteil des untersuchten Datenbestandes als auch Hilfsmittel für die weitere Datenübergabe sein.

Engrampa kann beispielsweise eingesetzt werden, um:

- vorhandene Archivdateien zu öffnen,
- den Inhalt eines Archivs einzusehen,
- einzelne oder mehrere Dateien zu extrahieren,
- Dateien zu einem neuen Archiv zusammenzufassen,
- gerettete Datenbestände geordnet für die weitere Verarbeitung vorzubereiten,
- Protokolle, Konfigurationsdateien oder Untersuchungsergebnisse gemeinsam abzulegen.

Damit ergänzt Engrampa die Dateiverwaltung von Thunar um Funktionen für komprimierte beziehungsweise zusammengefasste Datenbestände.

Funktionen

Engrampa stellt eine grafische Oberfläche für die Arbeit mit Archivdateien bereit.

Zu den wesentlichen Funktionen gehören:

- Öffnen vorhandener Archive,
- Anzeigen des Archivinhalts,
- Extrahieren einzelner oder mehrerer Dateien,
- Extrahieren vollständiger Archive,
- Erstellen neuer Archive,
- Hinzufügen von Dateien zu unterstützten Archiven,

- Entfernen von Einträgen aus unterstützten Archiven,
- Arbeiten mit verschiedenen Archiv- und Kompressionsformaten.

Welche Formate und Funktionen tatsächlich verfügbar sind, hängt neben Engrampa selbst auch von den im System vorhandenen Archiv- und Kompressionswerkzeugen ab.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue kann folgendermaßen aussehen:

1. Archivdatei auf dem untersuchten Datenträger oder innerhalb eines geretteten Datenbestandes lokalisieren.
2. Bei gefährdeten Originaldatenträgern zunächst eine Sicherung beziehungsweise Arbeitskopie erstellen.
3. Archiv möglichst von der Arbeitskopie aus öffnen.
4. Inhalt des Archivs prüfen.
5. Benötigte Dateien auswählen.
6. Separates Zielverzeichnis für die Extraktion festlegen.
7. Dateien extrahieren.
8. Extrahierte Dateien anschließend kontrollieren und gegebenenfalls mit geeigneten Werkzeugen weiter untersuchen.

Soll ein neues Archiv für die Übergabe von Daten erstellt werden, sollten die einzuschließenden Dateien zuvor gezielt zusammengestellt und kontrolliert werden.

Typische Rescue-Szenarien

Öffnen eines geretteten Archivs

Wurde beispielsweise mit einem Datenrettungswerkzeug eine Archivdatei wiederhergestellt, kann Engrampa verwendet werden, um zu prüfen, ob sich das Archiv öffnen und sein Inhalt extrahieren lässt.

Eine erfolgreich wiederhergestellte Archivdatei ist nicht automatisch vollständig oder fehlerfrei. Beschädigte beziehungsweise unvollständige Archivstrukturen können dazu führen, dass nur Teile des Inhalts zugänglich sind.

Extraktion benötigter Dateien

Befinden sich relevante Dokumente, Konfigurationen oder andere Dateien innerhalb eines Archivs, können diese in ein separates Arbeitsverzeichnis extrahiert und anschließend unabhängig vom Archiv untersucht werden.

Zusammenstellung von Untersuchungsergebnissen

Protokolle, Screenshots, Prüfsummen, Konfigurationsdateien und andere Ergebnisse eines Rescue-Einsatzes können bei Bedarf in einem Archiv zusammengefasst werden.

Dadurch lassen sich zusammengehörige Dateien als gemeinsamer Datenbestand aufbewahren oder übertragen.

Vorbereitung einer Datenübergabe

Gerettete Dateien können zunächst mit Thunar organisiert und anschließend mit Engrampa zu einem Archiv zusammengefasst werden, sofern dies für die weitere Übergabe zweckmäßig ist.

Grenzen und Sicherheitshinweise

Engrampa ist **kein Datenrettungswerkzeug**.

Das Programm stellt keine gelöschten Dateien wieder her, rekonstruiert keine verlorenen Partitionen und erzeugt keine sektorweisen Sicherungsabbilder.

Ebenso gilt:

Ein Archiv ist kein Sicherheitscontainer.

Das Verpacken von Dateien in einem Archiv verändert deren Vertrauensstatus nicht. Dateien aus einem kompromittierten oder unbekannten System bleiben auch nach dem Archivieren potenziell nicht vertrauenswürdig.

Beim Öffnen fremder oder geretteter Archive sollte deshalb berücksichtigt werden, dass deren Inhalt beispielsweise ausführbare Dateien, Skripte, manipulierte Dokumente oder andere potenziell gefährliche Inhalte enthalten kann.

Das bloße Extrahieren solcher Dateien macht sie nicht sicher.

Bei unbekannten Datenbeständen sollten extrahierte Inhalte daher nicht unüberlegt ausgeführt oder mit unnötig privilegierten Anwendungen geöffnet werden.

Auch eine erfolgreiche Extraktion beweist nicht, dass die enthaltenen Dateien inhaltlich unverändert, vollständig oder vertrauenswürdig sind.

Technische Einordnung in JUMBO BLOCK Rescue

Engrampa übernimmt innerhalb von Block C die grafische Verwaltung von Archivdateien.

Die funktionale Trennung ist dabei eindeutig:

Block A – JB-Rescue
dient der Sicherung, Analyse und Wiederherstellung von Daten.

C.2 – Thunar File Manager
verwaltet zugängliche Dateien und Verzeichnisse.

C.3 – Catfish File Search
sucht gezielt innerhalb zugänglicher Datenbestände.

C.4 – Engrampa Archive Manager
öffnet, extrahiert, erstellt und verwaltet unterstützte Archivdateien.

Engrampa wird damit insbesondere nach einer erfolgreichen Sicherung oder Datenrettung relevant, wenn archivierte Inhalte untersucht, extrahiert oder gerettete Daten für die weitere Verarbeitung zusammengefasst werden sollen.

Zentrale Regel:

Original beziehungsweise Arbeitskopie erhalten – Archive kontrolliert öffnen und extrahieren – unbekannte Inhalte nicht unüberlegt ausführen – Archivierung niemals mit Sicherheitsprüfung oder Vertrauenswürdigkeit gleichsetzen.

C.5 – FeatherPad

Bereich: Text
Programm: FeatherPad
Baseline 20261001: 1.6.2 ff.

Zweck

FeatherPad ist ein leichtgewichtiger grafischer Plain-Text-Editor.

Innerhalb von JUMBO BLOCK Rescue dient FeatherPad insbesondere zum Anzeigen und Bearbeiten textbasierter Dateien, beispielsweise Konfigurationsdateien, Protokolle, Skripte, Listen oder einfache Notizen.

Das Programm ermöglicht damit die direkte Untersuchung und – sofern erforderlich – gezielte Änderung von Textdateien innerhalb der Rescue-Umgebung.

Einsatzgebiet im Rescue-System

Bei Diagnose- und Reparaturarbeiten müssen häufig textbasierte Dateien untersucht werden.

FeatherPad kann beispielsweise eingesetzt werden, um:

- Konfigurationsdateien zu öffnen und zu prüfen,
- Protokolldateien zu lesen,
- bestimmte Zeichenfolgen innerhalb einer Datei zu suchen,
- einzelne Konfigurationseinträge gezielt zu ändern,
- Skripte und andere textbasierte Dateien einzusehen,
- Untersuchungsergebnisse oder technische Notizen festzuhalten,
- kopierte beziehungsweise gerettete Textdateien zu kontrollieren.

Durch die Unterstützung mehrerer Tabs können mehrere Dateien gleichzeitig geöffnet und miteinander verglichen beziehungsweise nacheinander bearbeitet werden.

Funktionen

Zu den für Rescue-Arbeiten relevanten Funktionen von FeatherPad gehören unter anderem:

- Öffnen und Speichern von Textdateien,
- Arbeiten mit mehreren Dateien über Tabs,
- Suchen und Ersetzen,
- Hervorhebung gefundener Suchbegriffe,
- Anzeige von Zeilennummern,
- Sprung zu einer bestimmten Zeile,

- Syntaxhervorhebung für verbreitete Programmiersprachen,
- Spaltenauswahl,
- Sitzungsverwaltung,
- Auto-Speichern,
- Textvergrößerung und -verkleinerung,
- Drucken.

FeatherPad ist dabei bewusst ein Texteditor und kein vollständiges Office- oder Dokumentenverarbeitungssystem.

Bedienung und Arbeitsablauf

Ein typischer Rescue-Arbeitsablauf kann folgendermaßen aussehen:

1. Betroffene Datei eindeutig identifizieren.
2. Feststellen, ob die Datei lediglich untersucht oder tatsächlich verändert werden soll.
3. Bei wichtigen Originaldateien zunächst eine Kopie erstellen.
4. Datei mit FeatherPad öffnen.
5. Inhalt prüfen beziehungsweise benötigte Stelle über die Suchfunktion lokalisieren.
6. Erforderliche Änderungen möglichst gezielt durchführen.
7. Geänderte Datei speichern.
8. Inhalt und Auswirkungen der Änderung anschließend kontrollieren.

Soll eine Konfigurationsdatei eines nicht mehr startfähigen Systems bearbeitet werden, ist besonders darauf zu achten, dass tatsächlich die Datei des gewünschten Zielsystems geöffnet wurde und nicht eine gleichnamige Datei des laufenden Rescue-Systems.

Typische Rescue-Szenarien

Prüfung von Konfigurationsdateien

Textbasierte System- und Anwendungskonfigurationen können geöffnet und auf fehlerhafte oder auffällige Einträge untersucht werden.

Gezielte Konfigurationsänderung

Ist die Ursache eines Problems eindeutig identifiziert, kann eine entsprechende Textdatei auf einer Arbeitskopie beziehungsweise im eingebundenen Zielsystem gezielt bearbeitet werden.

Vor einer Änderung sollte die ursprüngliche Datei erhalten bleiben.

Auswertung von Protokolldateien

Vorhandene Log- und Protokolldateien können geöffnet und mithilfe der Suchfunktion nach relevanten Begriffen, Meldungen oder Einträgen durchsucht werden.

Untersuchung geretteter Textdateien

Mit den Werkzeugen aus Block A wiederhergestellte Text-, Konfigurations- oder Skriptdateien können mit FeatherPad geöffnet und auf verwertbare Inhalte geprüft werden.

Dokumentation eines Rescue-Einsatzes

FeatherPad kann außerdem verwendet werden, um einfache technische Notizen, Befehlsfolgen, Dateilisten oder Untersuchungsergebnisse als Textdateien festzuhalten.

Grenzen und Sicherheitshinweise

FeatherPad ist **kein Datenrettungs-, Analyse- oder Sicherheitswerkzeug**.

Der Editor stellt keine gelöschten Dateien wieder her, repariert keine Dateisysteme und beurteilt nicht, ob der Inhalt einer geöffneten Datei korrekt, vertrauenswürdig oder ungefährlich ist.

Beim Bearbeiten von System- und Konfigurationsdateien ist besondere Vorsicht erforderlich.

Bereits kleine Änderungen können dazu führen, dass ein Betriebssystem, ein Dienst oder eine Anwendung anschließend nicht mehr ordnungsgemäß funktioniert.

Deshalb gilt:

Wichtige Originaldateien vor einer Änderung sichern und nur gezielte, nachvollziehbare Änderungen durchführen.

Bei Dateien aus einem kompromittierten System bleibt deren Herkunft ebenfalls zu berücksichtigen. Das Öffnen oder Speichern einer Datei mit FeatherPad bestätigt weder deren Integrität noch ihre Vertrauenswürdigkeit.

Automatische Speicherfunktionen sind bei Arbeiten an Originaldateien besonders zu beachten, da unbeabsichtigte Änderungen dadurch dauerhaft in die geöffnete Datei geschrieben werden können.

Technische Einordnung in JUMBO BLOCK Rescue

FeatherPad übernimmt innerhalb von Block C die Bearbeitung einfacher textbasierter Inhalte.

Die funktionale Trennung ist dabei klar:

C.2 – Thunar File Manager
verwaltet Dateien und Verzeichnisse.

C.3 – Catfish File Search
findet Dateien innerhalb zugänglicher Datenbestände.

C.4 – Engrampa Archive Manager
verwaltet und extrahiert Archivdateien.

C.5 – FeatherPad
zeigt textbasierte Dateien an und ermöglicht deren gezielte Bearbeitung.

FeatherPad ist damit insbesondere bei der Untersuchung von Konfigurationen und Protokollen sowie bei kontrollierten manuellen Änderungen an textbasierten Dateien ein praktisches Werkzeug innerhalb der Rescue-Umgebung.

Zentrale Regel:

Datei und Zielsystem eindeutig identifizieren – wichtige Originale vor Änderungen sichern – nur gezielte und nachvollziehbare Änderungen durchführen – gespeichertes Ergebnis anschließend kontrollieren.

C.6 – GtkHash

Bereich: Integrität
Programm: GtkHash
Baseline 20261001: 1.5 ff.

Zweck

GtkHash ist ein grafisches Werkzeug zur Berechnung und Überprüfung von kryptografischen Hashwerten beziehungsweise Prüfsummen von Dateien.

Innerhalb von JUMBO BLOCK Rescue dient GtkHash insbesondere dazu, Dateien vor und nach einer Übertragung miteinander zu vergleichen, vorhandene Referenzprüfsummen zu überprüfen und den unveränderten Zustand gesicherter oder geretteter Dateien nachvollziehbar zu dokumentieren.

Einsatzgebiet im Rescue-System

Bei Rescue-Arbeiten werden Dateien häufig von einem Datenträger auf ein anderes Medium, in ein Arbeitsverzeichnis oder in einen gesicherten Datenbestand übertragen.

GtkHash kann dabei beispielsweise eingesetzt werden, um:

- Hashwerte einzelner Dateien zu berechnen,
- Dateien anhand ihrer Hashwerte miteinander zu vergleichen,
- eine Datei gegen einen bekannten Referenzwert zu prüfen,
- vorhandene Prüfsummendateien zu überprüfen,
- die unveränderte Übertragung wichtiger Dateien zu kontrollieren,
- Hashwerte geretteter Dateien für eine technische Dokumentation festzuhalten.

Damit ergänzt GtkHash die Datei- und Übertragungswerkzeuge von JUMBO BLOCK Rescue um eine Möglichkeit zur Integritätskontrolle.

Funktionen

GtkHash unterstützt zahlreiche Hashverfahren. Dazu gehören unter anderem:

- **MD5**
- **SHA-1**
- **SHA-2**, darunter SHA-256 und SHA-512
- **SHA-3**
- **BLAKE2**

GtkHash unterstützt außerdem die Überprüfung von Prüfsummendateien, parallele beziehungsweise mehrfädige Hashberechnung und HMAC-Verfahren.

Welche Hashfunktionen im konkreten System zur Verfügung stehen, kann von den beim Erstellen des Programms verwendeten Bibliotheken und Optionen abhängen.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue besteht aus folgenden Schritten:

1. Zu prüfende Datei eindeutig identifizieren.
2. GtkHash starten.
3. Datei auswählen.
4. Gewünschtes Hashverfahren auswählen.
5. Hashwert berechnen lassen.
6. Ergebnis dokumentieren beziehungsweise mit einem vorhandenen Referenzwert vergleichen.
7. Nach einem Kopier- oder Übertragungsvorgang den Hashwert der Zieldatei erneut berechnen.
8. Ausgangs- und Zielwert exakt vergleichen.

Sind die mit demselben Verfahren berechneten Hashwerte identisch, besteht eine sehr starke technische Grundlage dafür, dass die verglichenen Dateiinhalte identisch sind.

Unterschiedliche Hashwerte zeigen dagegen, dass die verglichenen Daten nicht identisch sind.

Typische Rescue-Szenarien

Kontrolle einer Dateiübertragung

Eine wichtige gerettete Datei wird von einem Arbeitsdatenträger auf ein separates Sicherungsmedium kopiert.

Vor beziehungsweise nach der Übertragung können Hashwerte berechnet und miteinander verglichen werden. Dadurch lässt sich kontrollieren, ob sich der Dateiinhalt während des Kopiervorgangs verändert hat.

Dokumentation geretteter Dateien

Für besonders relevante Dateien können Hashwerte dokumentiert werden. Spätere Prüfungen ermöglichen dann festzustellen, ob der jeweilige Dateiinhalt gegenüber diesem dokumentierten Zustand unverändert geblieben ist.

Prüfung gegen eine veröffentlichte Prüfsumme

Liegt für eine Datei ein vertrauenswürdig bezogener Referenzwert vor, kann der mit GtkHash berechnete Wert damit verglichen werden.

Dies ist beispielsweise bei heruntergeladenen Abbildern, Softwarepaketen oder anderen Dateien sinnvoll, für die der Herausgeber eine Prüfsumme veröffentlicht hat.

Prüfung eines geretteten Datenbestandes

Nach einer erfolgreichen Datenrettung können ausgewählte Dateien mit Hashwerten versehen beziehungsweise gegen bereits vorhandene Referenzwerte geprüft werden.

Dabei ist zwischen erfolgreicher Datenrettung und nachgewiesener Integrität zu unterscheiden: Ohne einen vertrauenswürdigen früheren Referenzwert lässt sich lediglich der aktuelle Zustand der geretteten Datei dokumentieren.

Grenzen und Sicherheitshinweise

Ein Hashwert beantwortet zunächst die Frage:

Sind die verglichenen Daten identisch beziehungsweise hat sich ihr Inhalt gegenüber einem bekannten Zustand verändert?

Er beantwortet nicht automatisch die Frage:

Ist diese Datei vertrauenswürdig oder ungefährlich?

Eine kompromittierte, manipulierte oder mit Schadcode versehene Datei besitzt ebenfalls einen eindeutig berechenbaren Hashwert.

Wird der Hashwert einer bereits manipulierten Datei dokumentiert, bestätigt eine spätere Übereinstimmung lediglich, dass sich diese Datei seit der Berechnung nicht verändert hat.

Auch eine veröffentlichte Prüfsumme ist nur so vertrauenswürdig wie die Quelle, aus der der Referenzwert stammt.

Für einen Herkunfts- oder Authentizitätsnachweis sind deshalb gegebenenfalls zusätzliche kryptografische Verfahren, beispielsweise digitale Signaturen, erforderlich.

Für neue Integritätsnachweise sollten geeignete moderne Hashverfahren wie SHA-256 oder SHA-512 gegenüber älteren Verfahren wie MD5 oder SHA-1 bevorzugt werden.

Technische Einordnung in JUMBO BLOCK Rescue

GtkHash übernimmt innerhalb von Block C die allgemeine Integritätskontrolle auf Dateiebene.

Dabei besteht eine wichtige Abgrenzung zu **B.6 – ISO Signature Verifier**:

B.6 – ISO Signature Verifier
ist auf die Integritäts- und Authentizitätsprüfung von ISO-Abbildern anhand geeigneter Signatur- und Prüfsummeninformationen ausgerichtet.

C.6 – GtkHash
ist ein allgemeines Werkzeug zur Berechnung und zum Vergleich von Hashwerten beliebiger Dateien.

Damit kann GtkHash an vielen Stellen eines Rescue-Arbeitsablaufs eingesetzt werden, insbesondere vor und nach Kopier-, Sicherungs- oder Übergabevorgängen.

GtkHash ersetzt jedoch weder eine Malware-Analyse noch einen kryptografischen Herkunftsnachweis.

Zentrale Regel:

Datei eindeutig identifizieren – geeignetes Hashverfahren verwenden – Referenzwert und berechneten Wert exakt vergleichen – Prüfsummenübereinstimmung als Integritätsnachweis, nicht automatisch als Vertrauens- oder Sicherheitsnachweis verstehen.

C.7 – Advanced Network Configuration

Bereich: Netzwerk
Programm: Advanced Network Configuration
Baseline 20261001: 1.36.0 ff.

Zweck

Advanced Network Configuration (`nm-connection-editor`) ist die grafische Oberfläche zur Verwaltung von Netzwerkverbindungen des NetworkManager.

Innerhalb von JUMBO BLOCK Rescue dient das Werkzeug dazu, die Netzwerkverbindung der Rescue-Umgebung gezielt einzurichten oder vorhandene Verbindungsprofile anzupassen.

Damit können die Voraussetzungen für Netzwerkdiagnosen, den Zugriff auf Netzwerkressourcen oder andere netzwerkbasierte Rescue-Aufgaben geschaffen werden.

Einsatzgebiet im Rescue-System

Eine funktionierende Netzwerkverbindung kann während eines Rescue-Einsatzes beispielsweise benötigt werden für:

- Zugriff auf andere Systeme im lokalen Netzwerk,
- Nutzung von SSH-Verbindungen,
- Einbindung entfernter Systeme über SSHFS,
- Netzwerkdiagnosen mit den Werkzeugen aus Block A,
- DNS- und Erreichbarkeitstests,
- kontrollierten Zugriff auf erforderliche Internetressourcen.

Advanced Network Configuration übernimmt dabei nicht die eigentliche Diagnose dieser Verbindungen, sondern deren **Konfiguration**.

Funktionen

Das Programm kann von NetworkManager gespeicherte Netzwerkverbindungen:

- anlegen,
- bearbeiten,
- entfernen.

Die dokumentierten Verbindungstypen umfassen unter anderem:

- Ethernet,
- WLAN,

- Bridge,
- Bond,
- VLAN.

Je nach Verbindungstyp stehen die jeweils zugehörigen Konfigurationsmöglichkeiten zur Verfügung.

Dazu gehören bei entsprechenden Netzwerkverbindungen insbesondere Einstellungen für **IPv4 und IPv6**.

Voraussetzung für das Anlegen, Ändern oder Entfernen der Verbindungen ist ein laufender NetworkManager.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue besteht aus folgenden Schritten:

1. Vorhandene Netzwerkumgebung und gewünschte Verbindung bestimmen.
2. **Advanced Network Configuration** öffnen.
3. Vorhandenes Verbindungsprofil auswählen oder ein neues Profil anlegen.
4. Benötigte Verbindungseinstellungen eintragen beziehungsweise kontrollieren.
5. Änderungen speichern.
6. Gewünschte Netzwerkverbindung herstellen.
7. Erreichbarkeit des benötigten Zielsystems beziehungsweise Netzwerkdienstes anschließend mit geeigneten Werkzeugen prüfen.

Bei manueller IP-Konfiguration müssen insbesondere IP-Adresse, Netzmaske beziehungsweise Präfix, Gateway und gegebenenfalls DNS-Konfiguration zur vorhandenen Netzwerkumgebung passen.

Typische Rescue-Szenarien

Manuelle Netzwerkkonfiguration

Steht keine geeignete automatische Netzwerkkonfiguration zur Verfügung, kann für JUMBO BLOCK Rescue ein entsprechendes Verbindungsprofil eingerichtet werden.

Dies ermöglicht anschließend den Zugriff auf benötigte Netzwerkressourcen.

Vorbereitung einer SSH-Verbindung

Bevor **A.27 – SSH** für den Zugriff auf ein entferntes System verwendet werden kann, muss JUMBO BLOCK Rescue über eine funktionierende Netzwerkverbindung zum Ziel verfügen.

Advanced Network Configuration kann die hierfür erforderliche lokale Netzwerkkonfiguration bereitstellen.

Vorbereitung von SSHFS

Soll mit **A.28 – SSHFS** ein entferntes Dateisystem eingebunden werden, muss zunächst die entsprechende Netzwerkverbindung zum SSH-Server bestehen.

Netzwerkdiagnose

Für Werkzeuge wie Nmap, Wireshark, TCPDump, ARP Scan, MTR oder Dig kann eine korrekt konfigurierte Netzwerkschnittstelle erforderlich sein.

Advanced Network Configuration schafft hierfür die lokale Konfigurationsgrundlage. Die eigentliche Analyse erfolgt anschließend mit den spezialisierten Werkzeugen aus Block A.

Arbeiten in getrennten Netzwerksegmenten

Muss JUMBO BLOCK Rescue in einer Umgebung mit vorgegebenen Adressen, Gateways oder anderen Netzwerkeinstellungen eingesetzt werden, können entsprechende Verbindungsprofile gezielt eingerichtet werden.

Grenzen und Sicherheitshinweise

Advanced Network Configuration ist **kein Netzwerkdiagnose- oder Sicherheitsanalysewerkzeug**.

Das Programm erkennt beispielsweise nicht selbstständig:

- kompromittierte Netzwerkgeräte,
- unerlaubte Netzwerkdienste,
- manipulierten Netzwerkverkehr,
- Schadsoftware,
- Angriffe innerhalb des Netzwerks.

Hierfür sind die entsprechenden Analyse- und Sicherheitswerkzeuge aus Block A vorgesehen.

Fehlerhafte Netzwerkeinstellungen können außerdem dazu führen, dass Netzwerkressourcen nicht mehr erreichbar sind oder Datenverkehr über eine unbeabsichtigte Verbindung geleitet wird.

Besondere Vorsicht ist bei unbekannten oder möglicherweise kompromittierten Netzwerken erforderlich.

Eine technisch funktionierende Netzwerkverbindung bedeutet nicht, dass das verwendete Netzwerk vertrauenswürdig ist.

Ist für eine Rescue-Aufgabe keine Netzwerkverbindung erforderlich, besteht auch keine technische Notwendigkeit, das Rescue-System mit einem Netzwerk zu verbinden.

Technische Einordnung in JUMBO BLOCK Rescue

Advanced Network Configuration bildet die grafische Konfigurationsebene für Netzwerkverbindungen der Rescue-Umgebung.

Die funktionale Trennung ist eindeutig:

C.7 – Advanced Network Configuration
konfiguriert die Netzwerkverbindungen von JUMBO BLOCK Rescue.

A.17 bis A.23 – Netzwerk-Werkzeuge
untersuchen Netzwerkstrukturen, Verbindungen, Datenverkehr, Routing und Namensauflösung.

A.27 – SSH und A.28 – SSHFS
nutzen eine vorhandene Netzwerkverbindung für den entfernten Zugriff beziehungsweise die Einbindung entfernter Dateisysteme.

Advanced Network Configuration schafft somit die technische Grundlage für zahlreiche netzwerkbasierte Rescue-Aufgaben, ohne selbst deren Analysefunktion zu übernehmen.

Zentrale Regel:

Nur benötigte Netzwerkverbindungen einrichten – Konfigurationswerte vor dem Speichern kontrollieren – unbekannte Netzwerke nicht automatisch als vertrauenswürdig behandeln – anschließend die tatsächlich benötigte Erreichbarkeit gezielt prüfen.

C.8 – Xfce Terminal

Bereich: System
Programm: Xfce Terminal
Baseline 20261001: 1.1.4 ff.

Zweck

Xfce Terminal ist ein grafischer Terminalemulator der Xfce-Arbeitsumgebung.

Innerhalb von JUMBO BLOCK Rescue stellt das Programm eine grafische Umgebung für die Ausführung von Kommandozeilenbefehlen, Systemwerkzeugen und terminalbasierten Rescue-Programmen bereit.

Das Terminal selbst führt dabei keine Datenrettung, Diagnose oder Reparatur durch. Es stellt die Arbeitsoberfläche bereit, innerhalb derer entsprechende Befehle und Programme ausgeführt werden können.

Einsatzgebiet im Rescue-System

Ein wesentlicher Teil der technischen Rescue-Werkzeuge kann über die Kommandozeile verwendet werden.

Das Xfce Terminal ermöglicht beispielsweise:

- Ausführung von Diagnosebefehlen,
- Start terminalbasierter Rescue-Werkzeuge,
- Arbeit mit Datenträger- und Dateisystembefehlen,
- Ausführung von Netzwerkbefehlen,
- Nutzung von SSH und SSHFS,
- Berechnung und Kontrolle technischer Informationen,
- Navigation innerhalb von Verzeichnisstrukturen,
- kontrollierte Ausführung administrativer Befehle,
- Dokumentation beziehungsweise Übernahme von Befehlsausgaben.

Damit stellt das Terminal eine zentrale Schnittstelle zwischen der grafischen Rescue-Umgebung und den Kommandozeilenwerkzeugen des Systems dar.

Funktionen

Zu den grundlegenden Funktionen des Xfce Terminals gehören unter anderem:

- Ausführung einer Shell innerhalb eines grafischen Fensters,
- mehrere Terminal-Tabs,

- mehrere Terminalfenster,
- Kopieren und Einfügen von Text,
- Suchfunktion innerhalb der Terminalausgabe,
- Anpassung der Darstellung,
- Vollbildmodus,
- Scrollback für zurückliegende Terminalausgaben.

Mehrere Tabs ermöglichen es beispielsweise, unterschiedliche Arbeitsschritte oder Diagnosewerkzeuge parallel geöffnet zu halten.

Bedienung und Arbeitsablauf

Ein typischer Rescue-Arbeitsablauf kann folgendermaßen aussehen:

1. Xfce Terminal öffnen.
2. Aktuelle Arbeitsumgebung beziehungsweise benötigtes Verzeichnis bestimmen.
3. Vor einer schreibenden Operation Quelle und Ziel eindeutig identifizieren.
4. Gewünschten Befehl vollständig eingeben.
5. Befehl vor der Ausführung nochmals kontrollieren.
6. Befehl ausführen.
7. Ausgabe und mögliche Fehlermeldungen vollständig prüfen.
8. Erst danach den nächsten Arbeitsschritt durchführen.

Bei komplexeren Befehlen sollte besonders auf Gerätenamen, Pfade, Optionen und Zielangaben geachtet werden.

Ein syntaktisch gültiger Befehl ist nicht automatisch der richtige Befehl für den beabsichtigten Datenträger oder das gewünschte Zielsystem.

Typische Rescue-Szenarien

Start von Kommandozeilenwerkzeugen

Zahlreiche Programme aus **Block A – JB-Rescue** können direkt im Terminal gestartet und mit gezielten Optionen verwendet werden.

Dazu gehören beispielsweise Werkzeuge für:

- Datenträgerdiagnose,
- Datenrettung,
- Netzwerkanalyse,
- Hardwarediagnose,

- Verschlüsselung,
- Remote-Zugriff.

Prüfung von Datenträgern und Dateisystemen

Geräte, Partitionen, Einhängepunkte und Dateisysteme können mit geeigneten Systembefehlen ermittelt werden, bevor weitere Rescue-Schritte erfolgen.

Dies ist insbesondere vor schreibenden oder destruktiven Operationen wichtig.

Remote-Arbeiten

A.27 – SSH wird typischerweise innerhalb eines Terminals verwendet.

Auch **A.28 – SSHFS** kann über die Kommandozeile eingerichtet und kontrolliert werden.

Auswertung technischer Informationen

Ausgaben von Diagnoseprogrammen können unmittelbar im Terminal gelesen und für weitere Arbeitsschritte verwendet werden.

Arbeiten mit administrativen Rechten

Bestimmte Rescue-Aufgaben benötigen erhöhte Berechtigungen.

Das Terminal ermöglicht die Ausführung entsprechender Befehle innerhalb der vorgesehenen Rechteverwaltung des Systems.

Erhöhte Berechtigungen sollten ausschließlich verwendet werden, wenn sie für den konkreten Arbeitsschritt erforderlich sind.

Grenzen und Sicherheitshinweise

Xfce Terminal ist **kein Rescue-Werkzeug im engeren Sinne**.

Die Wirkung einer Aktion wird durch den innerhalb des Terminals ausgeführten Befehl bestimmt.

Dadurch kann das Terminal sowohl für rein lesende Diagnosen als auch für tiefgreifende Veränderungen an Datenträgern und Systemen verwendet werden.

Besondere Vorsicht gilt deshalb bei Befehlen, die:

- Dateien löschen oder überschreiben,
- Datenträger direkt beschreiben,
- Partitionstabellen verändern,
- Dateisysteme erzeugen oder verändern,
- Datenträger ein- oder aushängen,
- Zugriffsrechte verändern,
- Systemkonfigurationen bearbeiten.

Bei administrativen Rechten können fehlerhafte Befehle weitreichende Auswirkungen haben.

Befehle aus unbekannten Quellen sollten daher nicht ungeprüft übernommen und ausgeführt werden.

Ebenso sollten Gerätebezeichnungen wie `/dev/sda`, `/dev/sdb` oder `/dev/nvme...` niemals allein aufgrund einer vermeintlich bekannten Reihenfolge angenommen werden. Die tatsächliche Zuordnung muss im jeweiligen System überprüft werden.

Technische Einordnung in JUMBO BLOCK Rescue

Xfce Terminal bildet die allgemeine grafische Kommandozeilenschnittstelle der Rescue-Umgebung.

Die funktionale Trennung ist dabei wichtig:

C.8 – Xfce Terminal
stellt die Terminalumgebung bereit.

Die Shell
interpretiert die eingegebenen Befehle.

Die jeweiligen Rescue- und Systemprogramme
führen die eigentlichen Diagnose-, Sicherungs-, Analyse- oder Reparaturfunktionen aus.

Damit ist Xfce Terminal kein spezialisiertes Rescue-Werkzeug, aber eine wichtige Arbeitsoberfläche für zahlreiche technische Funktionen von JUMBO BLOCK Rescue.

Gerade bei schreibenden Datenträgeroperationen gilt dabei dasselbe Grundprinzip wie im gesamten Rescue-System:

Zuerst identifizieren und diagnostizieren – Originale möglichst unverändert erhalten – bei gefährdeten Datenträgern zuerst sichern – schreibende Befehle nur bewusst und nach eindeutiger Prüfung von Quelle und Ziel ausführen.

Zentrale Regel:

Befehl verstehen – Quelle und Ziel prüfen – benötigte Rechte bewusst einsetzen – erst danach ausführen – Ergebnis und Fehlermeldungen kontrollieren.

C.9 – Screenshot

Bereich: Dokumentation
Programm: Screenshot / Xfce4 Screenshooter
Baseline 20261001: 1.11.1 ff.

Zweck

Screenshot (`xfce4-screenshooter`) ist das Bildschirmaufnahme-Werkzeug der Xfce-Arbeitsumgebung.

Innerhalb von JUMBO BLOCK Rescue dient das Programm zur visuellen Dokumentation von Bildschirmzuständen, Fehlermeldungen, Diagnoseergebnissen, Programmeinstellungen und anderen während eines Rescue-Einsatzes angezeigten Informationen.

Damit können relevante Zustände ergänzend zu Protokollen, Textdateien und technischen Prüfergebnissen bildlich festgehalten werden.

Einsatzgebiet im Rescue-System

Während einer Diagnose oder Datenrettung können Informationen auf dem Bildschirm erscheinen, die für die spätere Dokumentation oder Nachvollziehbarkeit des Arbeitsablaufs relevant sind.

Screenshot kann beispielsweise eingesetzt werden, um:

- Fehlermeldungen festzuhalten,
- Diagnoseergebnisse zu dokumentieren,
- Programmeinstellungen zu erfassen,
- Datenträger- oder Partitionsdarstellungen zu sichern,
- Netzwerk- und Hardwareinformationen bildlich festzuhalten,
- Zwischenergebnisse eines Rescue-Vorgangs zu dokumentieren,
- Zustände vor und nach einer Maßnahme gegenüberzustellen.

Die Bildschirmaufnahme ergänzt damit die technische Dokumentation eines Rescue-Einsatzes.

Funktionen

Xfce4 Screenshooter ermöglicht Bildschirmaufnahmen unterschiedlicher Bereiche.

Dazu gehören insbesondere:

- Aufnahme des gesamten Bildschirms,
- Aufnahme des aktiven Fensters,

- Aufnahme eines ausgewählten Bildschirmbereichs,
- zeitverzögerte Aufnahme,
- Speichern der erzeugten Bildschirmaufnahme.

Welche Aufnahmeart verwendet wird, hängt davon ab, welche Information dokumentiert werden soll.

Für technische Dokumentationen ist häufig die Aufnahme eines einzelnen Fensters oder eines gezielt ausgewählten Bereichs ausreichend.

Bedienung und Arbeitsablauf

Ein typischer Arbeitsablauf innerhalb von JUMBO BLOCK Rescue besteht aus folgenden Schritten:

1. Zu dokumentierenden Zustand auf dem Bildschirm herstellen.
2. Prüfen, welche Informationen tatsächlich für die Dokumentation benötigt werden.
3. Darauf achten, dass keine unnötigen vertraulichen oder personenbezogenen Informationen sichtbar sind.
4. Screenshot starten.
5. Gesamten Bildschirm, Fenster oder gewünschten Bereich auswählen.
6. Aufnahme erstellen.
7. Datei in einem geeigneten Arbeits- oder Dokumentationsverzeichnis speichern.
8. Aufnahme anschließend auf Lesbarkeit und Vollständigkeit kontrollieren.
9. Bei Bedarf gemeinsam mit Protokollen, Prüfsummen oder weiteren Untersuchungsergebnissen sichern.

Bei mehreren Aufnahmen empfiehlt sich eine eindeutige und nachvollziehbare Dateibenennung.

Typische Rescue-Szenarien

Dokumentation einer Fehlermeldung

Eine während des Systemstarts, einer Diagnose oder eines Programmlaufs angezeigte Fehlermeldung kann unmittelbar als Bildschirmaufnahme festgehalten werden.

Dadurch bleibt die sichtbare Meldung auch nach dem Schließen des Programms erhalten.

Dokumentation eines Datenträgerzustands

Darstellungen aus GParted, GNOME Disks oder anderen grafischen Diagnosewerkzeugen können als ergänzende visuelle Dokumentation gespeichert werden.

Dokumentation von Einstellungen

Vor einer Änderung wichtiger Einstellungen kann der Ausgangszustand als Screenshot festgehalten werden.

Nach der Änderung kann eine weitere Aufnahme den anschließend sichtbaren Zustand dokumentieren.

Dokumentation von Analyseergebnissen

Grafisch dargestellte Ergebnisse oder relevante Terminalausgaben können bei Bedarf zusätzlich als Bildschirmaufnahme gesichert werden.

Bei umfangreichen Terminalausgaben oder technischen Messwerten sollten jedoch die eigentlichen Text- beziehungsweise Protokolldaten bevorzugt erhalten werden.

Übergabe eines Rescue-Vorgangs

Screenshots können gemeinsam mit Protokollen, Prüfsummen, geretteten Konfigurationsdateien und anderen Untersuchungsergebnissen Bestandteil einer technischen Übergabedokumentation sein.

Grenzen und Sicherheitshinweise

Ein Screenshot bildet ausschließlich den **sichtbaren Bildschirmzustand zum Zeitpunkt der Aufnahme** ab.

Er beweist nicht automatisch:

- die technische Ursache eines dargestellten Fehlers,
- die Vollständigkeit einer Diagnose,
- die Integrität eines untersuchten Systems,
- die Authentizität der dargestellten Daten,
- den unveränderten Zustand eines Datenträgers oder einer Datei.

Eine Bildschirmaufnahme ersetzt deshalb keine technischen Originaldaten.

Sind beispielsweise Protokolldateien, Diagnoseausgaben oder Konfigurationsdateien verfügbar, sollten diese zusätzlich beziehungsweise vorzugsweise in ihrer ursprünglichen Form gesichert werden.

Für Integritätskontrollen können ergänzend Hashwerte mit **C.6 – GtkHash** dokumentiert werden.

Besondere Aufmerksamkeit ist außerdem auf vertrauliche Informationen zu richten.

Screenshots können beispielsweise enthalten:

- Benutzernamen,
- Dateinamen und Verzeichnisstrukturen,

- IP-Adressen,
- Systeminformationen,
- personenbezogene Daten,
- Inhalte geöffneter Dokumente,
- andere nicht für die Weitergabe bestimmte Informationen.

Vor einer Weitergabe sollte deshalb geprüft werden, welche Informationen tatsächlich auf der Aufnahme sichtbar sind.

Technische Einordnung in JUMBO BLOCK Rescue

Screenshot übernimmt innerhalb von Block C die visuelle Dokumentation der Rescue-Arbeitsumgebung.

Die funktionale Trennung ist dabei eindeutig:

Technische Rescue- und Diagnosewerkzeuge
erzeugen die eigentlichen Untersuchungsergebnisse.

C.5 – FeatherPad
kann für textbasierte Notizen und Dokumentationen verwendet werden.

C.6 – GtkHash
ermöglicht die Integritätskontrolle von Dateien.

C.9 – Screenshot
hält einen sichtbaren Bildschirmzustand als Bilddatei fest.

Damit stellt Screenshot ein ergänzendes Dokumentationswerkzeug dar. Es ist besonders hilfreich, wenn ein grafisch dargestellter Zustand oder eine nur vorübergehend sichtbare Meldung erhalten werden soll.

Ein Screenshot sollte jedoch stets als **ergänzende visuelle Dokumentation** verstanden werden und nicht als Ersatz für die zugrunde liegenden technischen Daten.

Zentrale Regel:

Nur relevante Informationen erfassen – vertrauliche Bildschirminhalte beachten – technische Originaldaten zusätzlich erhalten – Screenshot als ergänzende Dokumentation und nicht als alleinigen technischen Nachweis verwenden.

Block D

Neben den spezialisierten Rescue-Werkzeugen aus **Block A – JB-Rescue**, den ausgewählten MX-Werkzeugen aus **Block B** und den allgemeinen Arbeitswerkzeugen aus **Block C – Zubehör** enthält JUMBO BLOCK Rescue die vollständige **JUMBO BLOCK Engineering Suite**.

Die Engineering Suite stellt die für Planung, Berechnung, Projektbearbeitung und Dokumentenarbeit vorgesehenen JUMBO BLOCK Programme innerhalb der lokalen Rescue-Umgebung bereit. Dazu gehören insbesondere Werkzeuge für Hydrologie, geometrische Planung, Kosten, Statik, Temperatur, Inspektion, Projektmanagement und Ausschreibung sowie ergänzende Programme für Berechnungen, Einheiten, Tabellen, Texte, Dokumente, PDF-Dateien, Archive und Dateiverwaltung.

Damit dient JUMBO BLOCK Rescue nicht ausschließlich der technischen Rettung ausgefallener Systeme. Die integrierte Engineering Suite ermöglicht zugleich, wesentliche Arbeitsprozesse unabhängig von einem ausgefallenen Arbeitsplatzsystem unmittelbar innerhalb der Rescue-Umgebung fortzuführen.

Die Programme der Engineering Suite werden in diesem Handbuch bewusst kompakt beschrieben. Für die vollständige Bedienung, sämtliche Funktionen und programmspezifischen technischen Hinweise stehen die jeweiligen separaten **JUMBO BLOCK Handbücher** zur Verfügung.

Programmgruppen

20 Programme der JUMBO BLOCK Engineering Suite:

D.1 – Hydro
Hydrologische Vorplanung für Regenereignisse, Rückhaltung, Speichervolumen und Brauchwassernutzung.

D.2 – Plan
Geometrische Projektplanung, Modulanzahl, Flächen, Volumen und Mengenermittlung.

D.3 – Budget
Projektkalkulation für Material, Transport, Montage und Gesamtübersicht.

D.4 – Static
Technische Bemessung und FE-Berechnung von Last- und Einbausituationen.

D.5 – Temp
Thermische Infrastruktur-Bilanz zur Bewertung von Wärmespeicherung, Temperaturentwicklung, Standortentlastung und thermischer Infrastruktur.

D.6 – Inspect
Lokales Wartungs- und Inspektionsbuch für JUMBO BLOCK Anlagen mit Prüfkatalog, Inspektionen, Mess- und Grenzwerten, Fotodokumentation, Mängeln und Berichten.

D.7 – Project
Projektmanagement für JUMBO BLOCK Engineering-Projekte und Machbarkeitsstudien einschließlich Verantwortlichkeiten, Terminen, Partnern und Dokumenten.

D.8 – Tender
Lokales Werkzeug zur Erstellung und Bearbeitung von GAEB DA XML Leistungsverzeichnissen der Datenphasen X80 bis X84.

D.9 – Calc
Universeller Engineering-Rechner für allgemeine, geometrische, wissenschaftliche und JUMBO BLOCK spezifische Berechnungen.

D.10 – Units
Offline-Einheitenumrechner für Engineering, Physik und technische Anwendungen.

D.11 – Note
Lokales Notizprogramm für Notizen, Suche, Import und Export sowie Verknüpfungen zu Webseiten und Dokumenten.

D.12 – Tables
Offline-Tabellenkalkulation mit Formel-Engine, Mehrblatt-Unterstützung, Zellformatierung, Diagrammen und verschiedenen Exportmöglichkeiten.

D.13 – Writer
Resilienter Offline-Writer zur vollständig lokalen Erstellung und Bearbeitung von Dokumenten.

D.14 – Reader
Offline-Reader für E-Books und technische Dokumente in verschiedenen Datei- und Dokumentformaten.

D.15 – Text
Offline-Editor für Text- und Quellcodedateien mit Syntaxhervorhebung, Zeilennummern sowie Such- und Bearbeitungsfunktionen.

D.16 – PDF Merge
Lokales Werkzeug zum Zusammenführen mehrerer PDF-Dateien sowie zum Sortieren und Entfernen einzelner Seiten.

D.17 – PDF Printer
Lokales Werkzeug für den kontrollierten Ausdruck vorhandener PDF-Dateien mit Vorschau und Druckeinstellungen.

D.18 – Convert
Lokaler Konverter zur Umwandlung von PDF-Dateien in bearbeitbare Word-Dokumente.

D.19 – Archive
Offline-Werkzeug zum Erstellen und Entpacken unterstützter Archivformate.

D.20 – Files
Offline-Dateimanager für lokale Ordner, USB-Sticks und Laufwerke mit Dateioperationen, Vorschau und Suchfunktionen.

Die 20 Programme bilden gemeinsam die innerhalb von JUMBO BLOCK Rescue verfügbare **Engineering- und Arbeitsumgebung**.

Während **D.1 bis D.8** vor allem die fachlichen Engineering-, Anlagen- und Projektprozesse abdecken, stellen **D.9 bis D.20** allgemeine Berechnungs-, Dokumenten-, Datei- und Arbeitsfunktionen bereit.

Block D ermöglicht damit, nach dem Start von JUMBO BLOCK Rescue nicht nur Systeme und Daten zu untersuchen und zu retten, sondern wesentliche JUMBO BLOCK Engineering- und Dokumentenprozesse vollständig lokal weiterzuführen.

Zuordnung der Bereiche, Programme und Versionen

Nr.	Bereich	Programm	Baseline-Version
D.1	Hydrologie	Hydro	9.3.1 ff.
D.2	Planung	Plan	9.2.3 ff.
D.3	Kosten	Budget	9.2.2 ff.
D.4	Statik	Static	11.0.2 ff.
D.5	Temperatur	Temp	9.0.0 ff.
D.6	Inspektion	Inspect	1.0.0 ff.
D.7	Projektmanagement	Project	1.1.2 ff.
D.8	Ausschreibung	Tender	4.0.0 ff.
D.9	Berechnung	Calc	2.0.0 ff.
D.10	Einheiten	Units	2.0.0 ff.
D.11	Notizen	Note	1.0.0 ff.
D.12	Tabellen	Tables	2.0.0 ff.

Nr.	Bereich	Programm	Baseline-Version
D.13	Dokumente	Writer	10.1.0 ff.
D.14	Dokumente	Reader	1.0.4 ff.
D.15	Text / Code	Text	1.0.0 ff.
D.16	PDF	PDF Merge	1.0.0 ff.
D.17	PDF	PDF Printer	1.2.1 ff.
D.18	Konvertierung	Convert	0.3.2 ff.
D.19	Archive	Archive	1.1.1 ff.
D.20	Dateien	Files	1.0.0 ff.

D.1 – Hydro

Bereich: Hydrologie
Programm: JUMBO BLOCK Hydro
Baseline 20261001: 9.3.1 ff.

Zweck

JUMBO BLOCK Hydro ist das hydrologische Abschätzungswerkzeug der JUMBO BLOCK Engineering Suite.

Das Programm liefert eine erste technische Einschätzung zur urbanen Regenwasserrückhaltung und verbindet hydrologische Eingangsdaten unmittelbar mit der modularen JUMBO BLOCK Anlage.

Zu den wesentlichen Aufgaben gehören:

- Flächenanalyse,
- Bemessungsregen,
- Speichernachweis nach DWA-A 117,
- Ermittlung von Regen- und Speichervolumen,
- Berechnung des erforderlichen Modulbedarfs,
- Starkregen-Szenarien,
- Stadtstraßen-Modell,
- Projekt- und Notfallplanung.

Rolle innerhalb der Engineering Suite

Hydro bildet die **hydrologische Grundlage** der Engineering Suite.

Aus Projektfläche, Versiegelungsgrad, Abflussbeiwert und standortbezogenen Regendaten wird der erforderliche Speicherbedarf ermittelt und auf eine reale JUMBO BLOCK Modulanlage übertragen.

Hydro kann eigenständig verwendet werden, ist gleichzeitig aber Bestandteil des gemeinsamen Datenmodells der Engineering Suite.

Die weitere geometrische und technische Bearbeitung eines Projekts kann anschließend in anderen Programmen der Suite fortgeführt werden.

Wesentliche Funktionen

Hydro unterstützt zwei voneinander unabhängige hydrologische Betrachtungen.

Das vereinfachte Verfahren ermittelt aus wirksamer Fläche, Regenhöhe, Standortfaktor und Abflussbeiwert das Regenvolumen und daraus den erforderlichen Modulbedarf.

Zusätzlich führt Hydro einen vergleichenden **DWA-A 117-Speichernachweis** über fünf Dauerstufen durch. Dabei werden Drosselabfluss und gegebenenfalls Versickerungsleistung berücksichtigt und die Dauerstufe mit dem größten erforderlichen Speicherbedarf als maßgebend bestimmt.

Die Anlagenberechnung überträgt den ermittelten Speicherbedarf auf JUMBO BLOCK Module. Das Nettovolumen des Standardmoduls ist dabei mit **13,9 m³** hinterlegt.

Weitere Funktionen umfassen unter anderem:

- Projektübersicht,
- Starkregen-Simulation,
- Stadtstraßen-Modell,
- Notfallplanung,
- Infrastruktur-Bereitschaft,
- Hydrological Report,
- Deutsch-/Englisch-Umschaltung,
- kontextbezogene fachliche Hilfen.

Daten und Datenaustausch

Hydro arbeitet mit der gemeinsamen Projektdatei:

project.dbm

Diese Projektdatei wird auch von anderen Programmen der JUMBO BLOCK Engineering Suite verwendet.

Beim Export wird ein aktualisierter **SHA-256-Hash** über den gespeicherten Datensatz erzeugt und gemeinsam mit den Projektdaten gespeichert.

Der Dateiname folgt dem Schema:

JJJJMMTT_HHMMSS_project.dbm

Beim Import prüft Hydro den gespeicherten Hash und übernimmt die für Hydro relevanten Daten aus der gemeinsamen Projektdatei.

Dazu gehören insbesondere:

- Projektdaten,
- Flächenanalyse,
- Bemessungsregen,
- Stadtstraßen-Modell,
- DWA-A 117-Eingaben,
- zugehörige Orientierungswerte,

- verfügbare gemeinsame Angaben zur geplanten Modulanzahl.

Eine von **Plan** gesetzte gemeinsame Modulanzahl wird dabei bevorzugt übernommen.

Nach Import und Export erfolgt eine Neuberechnung, damit Ergebnisse und Bewertungen dem aktuellen Datenstand entsprechen.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Hydro arbeitet vollständig lokal im Browser.

Für die eigentliche Programmnutzung sind insbesondere nicht erforderlich:

- Cloud-Anbindung,
- Server,
- Benutzerkonto.

Es erfolgt außerdem keine automatische Standortzuordnung von Regendaten.

Die fachlichen Regendaten – insbesondere die verwendeten **KOSTRA-DWD-2020-Regenhöhen** – werden bewusst manuell als geprüfte standortbezogene Werte eingegeben.

Damit kann Hydro innerhalb von JUMBO BLOCK Rescue auch dann für die hydrologische Projektbearbeitung eingesetzt werden, wenn der reguläre Arbeitsplatz oder zentrale IT-Dienste nicht zur Verfügung stehen.

Ein bestehendes Projekt kann über seine `project.dbm` übernommen und innerhalb der lokalen Rescue-Umgebung weiterbearbeitet werden.

Neue Projekte können ebenfalls vollständig lokal begonnen und anschließend wieder als `project.dbm` gesichert werden.

Datensicherung

Hydro verwendet keine dauerhafte browserbasierte Zwischenspeicherung für den aktuellen Projektstand.

Der Bearbeitungsstand befindet sich während der Arbeit im Arbeitsspeicher der geöffneten Seite.

Beim Schließen oder Neuladen können nicht exportierte Änderungen deshalb verloren gehen.

Für die Arbeit innerhalb von JUMBO BLOCK Rescue gilt daher:

Projektstände regelmäßig als `project.dbm` exportieren und auf einem geeigneten Datenträger sichern.

Der SHA-256-Hash innerhalb der Projektdatei dient der Nachvollziehbarkeit von Veränderungen. Eine Hash-Abweichung wird beim Import angezeigt.

Der Hash verhindert jedoch weder den Import noch eine anschließende Berechnung.

Berichte

Über **Druck / PDF** erzeugt Hydro einen eigenständigen **Hydrological Report**.

Dieser enthält unter anderem:

- Projektinformationen,
- Bemessungsregen,
- Flächenanalyse,
- DWA-A 117-Speichernachweis,
- JUMBO BLOCK Anlagenbemessung,
- Starkregen-Simulation,
- Stadtstraßen-Modell,
- Zusammenfassung und Bewertung,
- Dokumenteninformationen und Hashwerte.

Vor der Berichtserstellung wird der aktuelle Projektstand vollständig neu berechnet.

Grenzen und Hinweise

Hydro ist ein Werkzeug für die hydrologische **Vorplanung und technische Orientierung**.

Das Programm ersetzt keine Genehmigungsplanung und keine wasserrechtlich erforderliche Fachplanung.

Insbesondere erfolgt keine automatische Ermittlung standortbezogener KOSTRA-Rasterwerte. Die verwendeten Regenhöhen müssen als geprüfte Fachwerte eingegeben werden.

Orientierungswerte für Drosselabfluss und Versickerung ersetzen weder die Abstimmung mit der zuständigen Behörde beziehungsweise kommunalen Entwässerungsvorgaben noch erforderliche standortbezogene Untersuchungen.

Das vereinfachte Verfahren und der DWA-A 117-Speichernachweis stellen unterschiedliche Berechnungswege dar und können deshalb unterschiedliche Speicheranforderungen ergeben. Beide Ergebnisse sind entsprechend ihrer jeweiligen Berechnungsgrundlage zu betrachten.

Technische Einordnung in JUMBO BLOCK Rescue

Hydro stellt innerhalb von Block D die **hydrologische Engineering-Funktion** bereit.

Während die Blöcke A bis C vor allem der Rettung, Untersuchung, Sicherung und technischen Bearbeitung von Systemen und Daten dienen, ermöglicht Hydro die Fortführung beziehungsweise Wiederaufnahme hydrologischer JUMBO BLOCK Projektarbeit innerhalb der lokalen Rescue-Umgebung.

Die gemeinsame `project.dbm` bildet dabei die Verbindung zu weiteren Programmen der Engineering Suite.

Hydro → hydrologische Grundlage und Speicherbedarf

Plan → geometrische Projektplanung

Static → technische und statische Bemessung

Budget → Projektkalkulation

Temp → Temperatur und thermische Infrastruktur-Bilanz

Damit bleibt die fachliche Projektkette auch innerhalb von JUMBO BLOCK Rescue verfügbar.

Weiterführende Dokumentation

Die vollständige Bedienung, Berechnungsgrundlagen, Eingabefelder, Standardwerte, Formeln, Berichte und weiteren technischen Hinweise sind im separaten Handbuch dokumentiert:

JUMBO BLOCK Hydro – Benutzerhandbuch · Version 9.3.1

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Hydro“ auf **jumboblock.org.**

D.2 – Plan

Bereich: Planung
Programm: JUMBO BLOCK Plan
Baseline 20261001: 9.2.3 ff.

Zweck

JUMBO BLOCK Plan ist das geometrische Vorplanungswerkzeug der JUMBO BLOCK Engineering Suite.

Das Programm übersetzt ein erforderliches Rückhaltevolumen – typischerweise aus **Hydro** übernommen – oder vorgegebene Anlagenabmessungen in ein konkretes modulares JUMBO BLOCK Anlagenlayout.

Dabei werden insbesondere ermittelt:

- Modulanzahl,
- Blockraster,
- reale Anlagenabmessungen,
- Netto-Speichervolumen,
- Bodenplatten,
- Seitenplatten,
- Abdeckplatten,
- Bauteilvolumen und Gewichte,
- Erdreichentnahme,
- Fugenlängen der SABA-Abdichtung.

Rolle innerhalb der Engineering Suite

Plan bildet die **geometrische Grundlage** der Engineering Suite.

Während Hydro den hydrologisch erforderlichen Speicherbedarf bestimmt, überführt Plan diesen Bedarf in eine konkrete Anlagengeometrie.

Ein typischer Arbeitsablauf innerhalb der Suite ist:

Hydro → Plan → Static → weitere Projektbearbeitung

Das in Plan erzeugte Anlagenlayout kann anschließend insbesondere für die statische Bemessung in **Static**, die Kostenermittlung in **Budget** und weitere Berechnungen in **Temp** verwendet werden.

Wesentliche Funktionen

Plan besitzt zwei grundlegende Planungsmodi.

Planung über Zielvolumen

Aus einem vorgegebenen Speichervolumen wird die erforderliche Mindestanzahl der JUMBO BLOCK Module ermittelt.

Anschließend schlägt Plan automatisch ein möglichst kompaktes Standardraster vor.

Das tatsächlich erzeugte Raster kann dabei eine zusätzliche Kapazitätsreserve gegenüber dem vorgegebenen Zielvolumen enthalten.

Planung über Anlagenmaß

Alternativ kann eine Anlage aus einer vorgegebenen Länge und Breite entwickelt werden.

Die Rasterermittlung erfolgt auf Grundlage des JUMBO BLOCK Modulrasters von:

2,5 × 2,5 × 2,5 m

Dabei werden vollständige 2,5-m-Rasterschritte berücksichtigt.

Manuelle Anpassung

Automatisch ermittelte Werte können für Sonderlayouts gezielt übersteuert werden.

Dies betrifft insbesondere:

- Blockanzahl,
- Bodenplatten,
- Seitenplatten,
- Abdeckplatten.

Sobald eine solche manuelle Übersteuerung aktiv ist, wird die Planung als **Sonderlayout** behandelt.

Auch die Dicken der Boden-, Seiten- und Abdeckplatten können innerhalb der im Programm vorgesehenen Grenzen angepasst werden.

Anlagenlayout

Das berechnete Raster wird grafisch dargestellt.

Dadurch können Modulzahl, Raster, Grundfläche, Speichervolumen und reale Anlagenabmessungen unmittelbar kontrolliert werden.

SABA-Abdichtung

Plan berücksichtigt die gewählte Ausführung der SABA-Fugenabdichtung und ermittelt daraus die erforderlichen Fugenlängen.

Die Berücksichtigung der Abdeckplattenfugen kann separat gesteuert werden.

Daten und Datenaustausch

Plan verwendet wie Hydro die gemeinsame Projektdatei:

project.dbm

Eine aus Hydro exportierte `project.dbm` kann in Plan übernommen werden.

Sind hydrologische Ergebnisse vorhanden, übernimmt Plan daraus das relevante Zielvolumen. Liegt bereits eine geplante Modulanzahl vor, kann auch diese in die weitere Planung einbezogen werden.

Plan kann außerdem einen zuvor gespeicherten eigenen Planungsstand wieder einlesen.

Beim Export werden die aktuellen Planungsdaten in die gemeinsame `project.dbm` geschrieben. Bereits vorhandene Daten anderer Programme der Engineering Suite bleiben dabei erhalten.

Damit dient die `project.dbm` als gemeinsame Projektbasis für die Weitergabe des Planungsstands innerhalb der Suite.

Datenintegrität

Jede von Plan exportierte `project.dbm` enthält einen **SHA-256-Hash** über den gespeicherten Datensatz.

Beim Import wird dieser Hash automatisch geprüft.

Der Planungsbericht enthält zusätzlich getrennte Hashwerte für:

- Planungseingaben,
- berechnete Ergebnisse.

Damit lässt sich der dokumentierte Berechnungsstand technisch nachvollziehen.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Plan läuft vollständig lokal im Browser.

Für die Programmnutzung sind insbesondere nicht erforderlich:

- Cloud-Anbindung,
- Server,
- Benutzerkonto.

Dadurch steht die geometrische Projektplanung unmittelbar innerhalb von JUMBO BLOCK Rescue zur Verfügung.

Ein vorhandenes Projekt kann über seine `project.dbm` übernommen und weiterbearbeitet werden. Ebenso kann innerhalb der Rescue-Umgebung eine neue Anlagenplanung begonnen werden.

Das ist insbesondere dann relevant, wenn ein regulärer Engineering-Arbeitsplatz nicht verfügbar ist, die Projektarbeit aber mit den vorhandenen Projektdaten fortgeführt werden muss.

Datensicherung

Plan verwendet keine dauerhafte browserbasierte Zwischenspeicherung des aktuellen Projektstands.

Nicht exportierte Änderungen können deshalb beim Schließen oder Neuladen der Seite verloren gehen.

Für die Arbeit innerhalb von JUMBO BLOCK Rescue gilt daher:

Aktuelle Planungsstände regelmäßig als `project.dbm` exportieren und auf einem geeigneten Datenträger sichern.

Dies gilt insbesondere vor dem Zurücksetzen einer Planung oder dem Beenden der Rescue-Umgebung.

Berichte

Über **Druck / PDF** erzeugt Plan einen eigenständigen Planungsbericht.

Dieser enthält unter anderem:

- Projektinformationen,
- Planungsmodus,
- Anlagenlayout,
- Bauteilinformationen,
- Fugeninformationen,
- theoretische Referenzwerte,
- Eingabe- und Ergebnis-Hash.

Vor der Berichtserstellung erfolgt eine vollständige Neuberechnung des aktuellen Planungsstands.

Grenzen und Hinweise

Plan dient der **technischen und geometrischen Vorplanung**.

Das Programm ersetzt ausdrücklich keine projektbezogene statische Bemessung.

Diese erfolgt in **Static**.

Auch die in Plan dargestellten theoretischen Referenzwerte der verfügbaren Plattenbetonklassen stellen keinen statischen Nachweis dar.

Kosten werden ebenfalls nicht in Plan berechnet. Die Kostenschätzung erfolgt innerhalb der Engineering Suite mit **Budget** auf Grundlage des erzeugten Anlagenlayouts.

Bei Grundstücken oder Anlagen mit stark von einem kompakten Standardraster abweichender Geometrie können manuelle Anpassungen beziehungsweise die Planung über Anlagenmaße erforderlich sein.

Technische Einordnung in JUMBO BLOCK Rescue

Plan stellt innerhalb von Block D die **geometrische Engineering-Funktion** bereit.

Die grundlegende Aufgabenteilung innerhalb der Engineering Suite lautet:

Hydro → hydrologischer Speicherbedarf

Plan → Modulanzahl, Raster, Anlagengeometrie und Mengen

Static → projektbezogene technische und statische Bemessung

Budget → Projektkalkulation

Temp → Temperatur und thermische Infrastruktur-Bilanz

Damit bildet Plan die Verbindung zwischen dem hydrologisch ermittelten Bedarf und der konkret geplanten JUMBO BLOCK Anlage.

Innerhalb von JUMBO BLOCK Rescue ermöglicht dies, vorhandene Planungsstände aus einer `project.dbm` wieder aufzunehmen oder die geometrische Projektplanung unabhängig von einem ausgefallenen regulären Arbeitsplatz fortzuführen.

Weiterführende Dokumentation

Die vollständige Bedienung, Planungsmodi, Eingabefelder, Rasterfindung, Sonderlayouts, SABA-Fugenlogik, Berechnungsgrundlagen, Berichte und weiteren technischen Hinweise sind im separaten Handbuch dokumentiert:

JUMBO BLOCK Plan – Benutzerhandbuch

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Plan“ auf **https://jumboblock.org/**.

D.3 – Budget

Bereich: Kosten
Programm: JUMBO BLOCK Budget
Baseline 20261001: 9.2.2 ff.

Zweck

JUMBO BLOCK Budget ist das Werkzeug zur überschlägigen Projektkalkulation innerhalb der JUMBO BLOCK Engineering Suite.

Das Programm verbindet die zuvor ermittelten Anlagen- und Bauteilmengen mit den für die Kalkulation hinterlegten Kostenansätzen und erstellt daraus eine strukturierte Kostenübersicht.

Berücksichtigt werden insbesondere:

- Anlagenkomponenten,
- Materialmengen,
- Bauteilmengen und -gewichte,
- Transport,
- Montage,
- weitere kalkulationsrelevante Positionen,
- Gesamtkosten des betrachteten Projektstands.

Budget dient damit der wirtschaftlichen Bewertung einer zuvor technisch beziehungsweise geometrisch bestimmten JUMBO BLOCK Anlage.

Rolle innerhalb der Engineering Suite

Budget bildet die **kaufmännische Kalkulationsebene** der Engineering Suite.

Die erforderlichen technischen Mengen werden nicht unabhängig in Budget neu geplant, sondern können aus den vorgelagerten Engineering-Programmen übernommen werden.

Dabei sind zwei unterschiedliche Datenstände möglich:

Plan → Budget
liefert eine Kalkulationsgrundlage auf Basis der geometrischen Vorplanung.

Static → Budget
ermöglicht eine realistischere Kalkulation auf Grundlage der statisch beziehungsweise technisch bestimmten Bauteile und Mengen.

Damit bleibt im Budget nachvollziehbar, auf welchem technischen Bearbeitungsstand die jeweilige Kostenschätzung basiert.

Wesentliche Funktionen

Budget strukturiert die Projektkosten in einzelne kalkulierbare Positionen.

Die Kalkulation verbindet dabei technische Mengen mit den jeweils angesetzten Kosten.

Zu den wesentlichen Informationen einer Position gehören – soweit für die jeweilige Position vorhanden – insbesondere:

- Beschreibung,
- Größe beziehungsweise Abmessung,
- Menge,
- Einzelgewicht,
- Positionsgewicht,
- Einheitspreis,
- Positionskosten.

Zusätzlich werden die relevanten Gesamtgewichte und Gesamtkosten zusammengeführt.

Damit lässt sich nicht nur der monetäre Projektumfang abschätzen, sondern auch die für Transport und Projektlogistik relevante Größenordnung der Bauteile nachvollziehen.

Daten und Datenaustausch

Budget ist in das gemeinsame Datenmodell der JUMBO BLOCK Engineering Suite eingebunden.

Als technische Ausgangsbasis können Projektdaten aus **Plan** beziehungsweise **Static** verwendet werden.

Die Herkunft der Daten ist dabei fachlich relevant:

Import aus Plan

Der Planungsstand stellt die geometrisch ermittelten Anlagen- und Bauteilmengen zur Verfügung.

Diese Variante eignet sich für eine frühe Budgetabschätzung, bevor eine weitergehende statische Bemessung vorliegt.

Import aus Static

Liegen bereits technisch beziehungsweise statisch bestimmte Bauteile und Mengen aus Static vor, können diese als Grundlage für eine realistischere Budgetbetrachtung verwendet werden.

Damit kann die Kalkulation mit zunehmendem technischen Bearbeitungsstand des Projekts fortgeschrieben werden.

Kalkulationsprinzip

Budget trennt grundsätzlich zwischen:

technischen Mengen
und

kaufmännischen Kostenansätzen.

Die Engineering-Programme liefern die geometrischen beziehungsweise technisch bestimmten Mengen.

Budget ordnet diesen Mengen die für die Kalkulation verwendeten Preisansätze zu und berechnet daraus die Positions- und Gesamtkosten.

Eine Änderung von Preisen verändert deshalb nicht die technische Anlagenplanung. Umgekehrt können Änderungen an Modulanzahl, Bauteilen oder Gewichten die Kalkulationsgrundlage verändern.

Diese Trennung ermöglicht es, technische und wirtschaftliche Änderungen nachvollziehbar voneinander zu behandeln.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Budget ist Bestandteil der lokal verfügbaren JUMBO BLOCK Engineering Suite.

Dadurch kann eine Projektkalkulation innerhalb von JUMBO BLOCK Rescue auch dann fortgeführt oder neu aufgebaut werden, wenn der reguläre Engineering-Arbeitsplatz oder zentrale IT-Dienste nicht verfügbar sind.

Bereits vorhandene Projektdaten können als Grundlage für die weitere Kalkulation dienen.

Dies ermöglicht beispielsweise:

- Wiederaufnahme einer unterbrochenen Projektkalkulation,
- überschlägige Kostenbewertung eines vorhandenen Planungsstands,
- Aktualisierung von Kostenansätzen,
- Erstellung einer neuen Kostenübersicht aus geretteten beziehungsweise vorhandenen Projektdaten,
- lokale Weiterarbeit während eines IT-Ausfalls.

Die technische Planung und die Kalkulation bleiben damit auch innerhalb der Rescue-Umgebung miteinander verbunden.

Berichte

Budget stellt die Kalkulation in einer strukturierten Kostenübersicht dar.

Der Bericht dokumentiert den zugrunde liegenden Projekt- und Kalkulationsstand und fasst die relevanten Positionen sowie die daraus resultierenden Kosten zusammen.

Dabei ist insbesondere die Herkunft der technischen Daten von Bedeutung, da eine Kalkulation auf Grundlage einer geometrischen Vorplanung aus Plan einen anderen technischen Bearbeitungsstand besitzt als eine Kalkulation auf Grundlage von Static.

Grenzen und Hinweise

Budget ist ein Werkzeug zur **Budget- und Kostenabschätzung**.

Die Qualität des Ergebnisses hängt unmittelbar von zwei Grundlagen ab:

- den übernommenen technischen Mengen,
- den verwendeten Kostenansätzen.

Preisänderungen, regionale Unterschiede, Lieferbedingungen, Transportentfernungen, projektspezifische Randbedingungen oder später geänderte technische Anforderungen können deshalb zu abweichenden tatsächlichen Projektkosten führen.

Eine Kalkulation aus **Plan** basiert auf einem früheren technischen Bearbeitungsstand als eine Kalkulation aus **Static**.

Diese beiden Kalkulationsgrundlagen sollten deshalb nicht ohne Berücksichtigung ihrer jeweiligen Datenbasis miteinander gleichgesetzt werden.

Budget ersetzt außerdem weder die geometrische Planung noch die statische Bemessung.

Diese Aufgaben verbleiben bei den dafür vorgesehenen Engineering-Programmen.

Technische Einordnung in JUMBO BLOCK Rescue

Budget stellt innerhalb von Block D die **wirtschaftliche Engineering-Funktion** bereit.

Die grundlegende Aufgabenteilung lautet:

Hydro → hydrologischer Speicherbedarf

Plan → Modulanzahl, Raster, Anlagengeometrie und Mengen

Static → technische und statische Bemessung

Budget → Kostenbewertung des jeweiligen technischen Projektstands

Temp → Temperatur und thermische Infrastruktur-Bilanz

Budget übernimmt damit die Ergebnisse der technischen Projektbearbeitung und übersetzt sie in eine wirtschaftliche Projektbetrachtung.

Innerhalb von JUMBO BLOCK Rescue kann dadurch nach einem Ausfall nicht nur die technische Planung weitergeführt, sondern auch deren finanzielle Größenordnung lokal bewertet und dokumentiert werden.

Weiterführende Dokumentation

Die vollständige Bedienung, Kalkulationsstruktur, Kostenpositionen, Importe, Berechnungen, Berichte und weiteren technischen Hinweise sind im separaten Handbuch dokumentiert:

JUMBO BLOCK Budget – Benutzerhandbuch · Version 9.2.2

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Budget“ auf **https://jumboblock.org/.**

D.4 – Static

Bereich: Statik
Programm: JUMBO BLOCK Static
Baseline 20261001: 11.0.2 ff.
Status: Engineering-Service / Informationsseite

Status innerhalb von JUMBO BLOCK Rescue

JUMBO BLOCK Static nimmt innerhalb der Engineering Suite eine Sonderstellung ein.

Die in JUMBO BLOCK Rescue enthaltene Anwendung ist eine **Informations- und Übergabeseite für den JUMBO BLOCK Engineering-Service**.

Das eigentliche Statik-Werkzeug und die darin enthaltenen Berechnungsmodelle sind **interne Engineering-Werkzeuge** und für Außenstehende nicht zugänglich.

Die auf JUMBO BLOCK Rescue vorhandene Static-Seite führt daher selbst keine statische Berechnung durch.

Sie erläutert:

- den Umfang der statischen Vorbemessung,
- die erforderlichen Eingangsdaten,
- den Ablauf der Projektübergabe,
- die betrachteten technischen Nachweise,
- die anschließend bereitgestellten Berichte.

Die Informationsseite selbst läuft vollständig lokal im Browser und überträgt keine Daten.

Zweck des Engineering-Service

JUMBO BLOCK Static dient der **statischen Vorbemessung, Bewehrungsplanung und Mengenermittlung für JUMBO BLOCK Anlagen**.

Aus der zuvor erstellten Anlagenplanung wird durch JUMBO BLOCK Engineering eine statische Vorbemessung einschließlich Bewehrung und Mengen entwickelt.

Grundlage ist die Projektdatei:

project.dbm

Die Projektdatei kann aus **JUMBO BLOCK Plan oder Budget** stammen. Die Berechnung selbst erfolgt anschließend mit dem internen Statik-Werkzeug von JUMBO BLOCK Engineering.

Ablauf

Der auf der Informationsseite beschriebene Ablauf besteht aus drei Schritten:

1. Planen

Die JUMBO BLOCK Anlage wird zunächst in **Plan oder Budget** vorbereitet und anschließend als `project.dbm` exportiert.

2. Projektdaten übergeben

Die `project.dbm` wird zusammen mit den für die statische Vorbemessung erforderlichen zusätzlichen Projektinformationen an JUMBO BLOCK Engineering übergeben.

3. Ergebnisse erhalten

JUMBO BLOCK Engineering berechnet die Anlage mit dem internen Statik-Werkzeug.

Als Ergebnis werden die zugehörigen technischen PDF-Berichte bereitgestellt.

Leistungsumfang des Engineering-Service

Die Anlage wird als Gesamtsystem betrachtet. Dabei wirken Module, Rahmen, Platten und Baugrund zusammen.

Abdeckplatte

Für die Abdeckplatte werden unter anderem Verkehrslasten von PKW bis Schwerlastverkehr, Überschüttung und Zusatzlasten betrachtet.

Die Informationsseite nennt hierbei insbesondere:

- Tragfähigkeit,
- Rissbreite,
- Durchbiegung,
- Ermüdung.

Als Lastsituationen werden unter anderem **SLW 60** und **F900** genannt.

Boden- und Seitenplatten

Bei Boden- und Seitenplatten werden unter anderem berücksichtigt:

- Erddruck,
- Wasserfüllung,
- Grundwasser.

Leerer und gefüllter Zustand werden getrennt betrachtet; der ungünstigere Zustand ist maßgebend.

Gesamtsystem

Module, umlaufender Rahmen und Baugrund werden gemeinsam als räumliches Modell der gesamten Anlage betrachtet.

Standsicherheit

Die statische Vorbemessung umfasst nach der Informationsseite:

- Grundbruch,
- Kippen,
- Gleiten,
- Sicherheit gegen Auftrieb,
- Sicherheit gegen Aufschwimmen bei Grundwasser.

Dauerhaftigkeit und Robustheit

Betrachtet werden unter anderem:

- Expositionsklasse,
- Betondeckung,
- Betonklasse,
- Lastumlagerungsfähigkeit der Anlage bei lokalem Ausfall.

Bewehrung und Optimierung

Der Engineering-Service umfasst außerdem die Ermittlung beziehungsweise Optimierung von:

- Plattendicken,
- Standardbewehrung,
- Rahmenbewehrung,
- Materialmengen,
- Gewichten,
- Bestellmengen.

Diese Punkte beschreiben den Leistungsumfang des internen Engineering-Service. Sie sind **keine öffentlich bedienbaren Berechnungsfunktionen der auf JUMBO BLOCK Rescue enthaltenen Informationsseite**.

Benötigte Projektdaten

Die Anlagengeometrie wird aus der `project.dbm` übernommen.

Für die Berechnung werden darüber hinaus projektbezogene Angaben benötigt.

Die Informationsseite nennt insbesondere:

- `project.dbm` aus Plan oder Budget mit Modulen, Raster und Plattenaufbau,
- Nutzung der Oberfläche,

- Verkehrslastklasse, beispielsweise PKW, LKW oder SLW 60,
- Material und Höhe der Überschüttung über der Abdeckplatte,
- Bodenart beziehungsweise zulässige Bodenpressung,
- vorhandenes Baugrundgutachten,
- Grundwassereinfluss und Grundwasserstand,
- Betriebsart – Rückhaltung mit Bodenplatte oder Versickerung ohne Bodenplatte,
- Umgebungsbedingungen für die Expositionsklasse, beispielsweise Tausalz und Frost,
- Projektangaben und Ansprechpartner für die Berichte.

Fehlende Werte werden nach der Beschreibung der Informationsseite gemeinsam mit dem Auftraggeber abgestimmt.

Ergebnisse und Berichte

Als Ergebnis des Engineering-Service werden drei aufeinander abgestimmte PDF-Berichte bereitgestellt.

Statik-Report

Der Statik-Report ist der vollständige technische Bericht mit Eingaben, Nachweisen und Ergebnissen.

Er ist insbesondere für Planer und Prüfer vorgesehen.

Produktionsausdruck

Der Produktionsausdruck enthält:

- Bewehrungspläne für Abdeckplatten,
- Bewehrungspläne für Seitenplatten,
- Bewehrungspläne für Bodenplatten,
- Rahmenbewehrung,
- Stahlliste mit Mengen und Gewichten.

Kunden-Report

Der Kunden-Report fasst die Ergebnisse verständlich für Bauherren, Kommunen und die Projektkommunikation zusammen.

Die Berichte tragen eine **Eingabe- und eine Ergebnis-Prüfsumme mittels SHA-256**. Dadurch kann der jeweilige Bericht dem berechneten Projektstand zugeordnet werden.

Grenzen

Die Berechnung wird ausdrücklich als:

statische Vorbemessung und Plausibilisierung des Engineerings

eingeordnet.

Für die Nutzung innerhalb von JUMBO BLOCK Rescue ist zusätzlich entscheidend:

Die integrierte Static-Seite führt selbst keine statischen Berechnungen durch.

Sie stellt weder das interne Berechnungsmodell noch eine öffentlich bedienbare Oberfläche für dessen Berechnungen bereit.

Eine vorhandene `project.dbm` kann innerhalb der übrigen Engineering Suite vorbereitet und gesichert werden. Für die eigentliche statische Vorbemessung muss sie jedoch zusammen mit den erforderlichen zusätzlichen Projektinformationen an JUMBO BLOCK Engineering übergeben werden.

Bedeutung innerhalb von JUMBO BLOCK Rescue

Static ist innerhalb von JUMBO BLOCK Rescue deshalb **nicht als autarkes Berechnungsprogramm**, sondern als Schnittstelle zwischen der lokal möglichen Projektvorbereitung und dem internen JUMBO BLOCK Engineering-Prozess einzuordnen.

Die Engineering-Kette ist an dieser Stelle bewusst geteilt:

Hydro → lokale hydrologische Vorplanung

Plan → lokale geometrische Projektplanung

Static → Informations- und Übergabeseite zum internen Engineering-Service

Die lokale Rescue-Umgebung ermöglicht damit, vorhandene Projektdaten zu öffnen, die Planung weiterzuführen und die für die statische Bearbeitung erforderliche `project.dbm` vorzubereiten beziehungsweise zu sichern.

Die eigentliche statische Vorbemessung bleibt dagegen dem internen JUMBO BLOCK Engineering-Werkzeug vorbehalten.

Diese Abgrenzung verhindert zugleich, dass die Verfügbarkeit der Static-Informationsseite innerhalb von JUMBO BLOCK Rescue mit der Verfügbarkeit des internen Statik-Werkzeugs verwechselt wird.

Weiterführende Informationen

Weitere Informationen zum Leistungsumfang und zur statischen Vorbemessung von JUMBO BLOCK Anlagen:

https://jumboblock.org/

D.5 – Temp

Bereich: Temperatur
Programm: JUMBO BLOCK Temp
Baseline 20261001: 9.0.0 ff.
Status: Engineering-Service / Informationsseite

Status innerhalb von JUMBO BLOCK Rescue

JUMBO BLOCK Temp nimmt innerhalb der Engineering Suite eine Sonderstellung ein.

Die in JUMBO BLOCK Rescue enthaltene Anwendung ist eine **Informations- und Übergabeseite für den JUMBO BLOCK Engineering-Service**.

Das eigentliche Werkzeug zur Berechnung der **Thermischen Infrastruktur-Bilanz (TIB)** und die darin enthaltenen Berechnungsmodelle sind interne Engineering-Werkzeuge und für Außenstehende nicht zugänglich.

Die auf JUMBO BLOCK Rescue vorhandene Temp-Seite führt daher selbst keine thermische Bilanzierung durch.

Sie erläutert:

- Zweck und Umfang der Thermischen Infrastruktur-Bilanz,
- den Ablauf der Projektübergabe,
- die betrachteten thermischen Größen,
- die benötigten Projektinformationen,
- die anschließend bereitgestellten Ergebnisse und Berichte.

Die Informationsseite selbst läuft vollständig lokal im Browser und überträgt keine Daten.

Zweck des Engineering-Service

JUMBO BLOCK Temp dient der Erstellung einer **Thermischen Infrastruktur-Bilanz (TIB)** für JUMBO BLOCK Anlagen.

Dabei wird der begehbare, wasserführende Infrastrukturraum unter der Oberfläche als thermisches System betrachtet.

Wasser und Beton bilden unter versiegelten Flächen eine thermische Speichermasse. Die TIB betrachtet diesen Zusammenhang im Hinblick auf Wärmespeicherung, Medientemperaturen, Standortentlastung und die Nutzung des Bauwerks als Infrastrukturraum.

Grundlage der Bearbeitung ist die Projektdatei:

project.dbm

Die Projektdatei kann aus **JUMBO BLOCK Plan oder Budget** stammen.

Auf dieser Grundlage erstellt JUMBO BLOCK Engineering die thermische Infrastruktur-Bilanz der Anlage.

Ablauf

Der auf der Informationsseite beschriebene Ablauf besteht aus drei Schritten:

1. Planen

Die JUMBO BLOCK Anlage wird zunächst in **Plan oder Budget** vorbereitet und anschließend als `project.dbm` exportiert.

2. Projektdaten übergeben

Die `project.dbm` wird zusammen mit den zusätzlich benötigten Angaben an JUMBO BLOCK Engineering übergeben.

3. Ergebnisse erhalten

JUMBO BLOCK Engineering bilanziert die Anlage mit dem internen Temp-Werkzeug.

Als Ergebnis wird ein **TIB-Bericht mit Kennzahlen und Diagrammen als PDF** bereitgestellt.

Leistungsumfang des Engineering-Service

Die Thermische Infrastruktur-Bilanz betrachtet verschiedene thermisch wirksame Bestandteile und Prozesse der Anlage.

Wärmespeicher

Als thermische Speichermassen werden berücksichtigt:

- Wasser,
- Modulbeton,
- Platten,
- Aufschüttung.

Die Betrachtung umfasst Wasserraum und Trockenraum sowie einen optionalen Flutbetrieb.

Wärmeabfuhr

Die Bilanz berücksichtigt:

- Verdunstung,
- Wärmeabgabe an das Erdreich,
- Konvektion an die Luft.

Wärmeeinträge

Als Wärmeeinträge können unter anderem berücksichtigt werden:

- Sonneneinstrahlung über die Oberfläche,
- Kabel,
- Abwärmeleitungen,
- Medienleitungen,
- technische Einrichtungen im Infrastrukturraum.

Medientemperaturen und Betrieb

Betrachtet werden Temperaturen von Luft und Wasser in Wasser- und Trockenraum sowie deren Bedeutung für Betrieb und Wartung.

Standortwirkung

Die TIB betrachtet die lokale Entlastung der Oberfläche und die Oberflächentemperatur mit und ohne JUMBO BLOCK.

Standort-TIB

Als weitere Kenngröße wird der thermische Kompensationsgrad in Bezug auf die Wärmeeinträge des Standorts und seiner technischen Anlagen betrachtet.

Diese Punkte beschreiben den Leistungsumfang des internen Engineering-Service. Sie sind **keine öffentlich bedienbaren Berechnungsfunktionen der auf JUMBO BLOCK Rescue enthaltenen Informationsseite**.

Benötigte Projektdaten

Die Bauwerksgeometrie wird aus der `project.dbm` übernommen.

Für die thermische Bilanz werden darüber hinaus projektspezifische Informationen benötigt.

Die Informationsseite nennt insbesondere:

- `project.dbm` aus Plan oder Budget mit Modulen und Plattenaufbau,
- Aufteilung in Wasserraum und Trockenraum,
- Wasserfüllung,
- gegebenenfalls Flutbetrieb,
- versiegelte Fläche des Gesamtstandorts,
- bekannte Wärmeeinträge des Standorts durch Gebäude und technische Anlagen,
- Anzahl, Länge und Verlustleistung geplanter Kabel,
- geplante Abwärme- und Medienleitungen,
- technische Einrichtungen und deren Leistung,
- Aufschüttung beziehungsweise Aufbau über der Abdeckplatte,

- gewünschtes Klimaprofil,
- gewünschten Betrachtungszeitraum.

Fehlende Werte werden nach der Beschreibung der Informationsseite gemeinsam mit dem Auftraggeber abgestimmt.

Ergebnisse und Berichte

Als Ergebnis des Engineering-Service wird ein TIB-Bericht bereitgestellt.

TIB-Bericht

Der Bericht enthält eine Kernbilanz mit:

- thermischer Reserve und Auslastung,
- Medientemperaturen,
- Betrieb und Wartung,
- Standortwirkung,
- Standort-TIB,
- Diagrammen,
- Ampelbewertung.

Detailbilanz

Die Detailbilanz stellt die einzelnen Positionen der Wärmebilanz für Planung und Prüfung zusammen.

Klimaprofile

Die Anlage kann für typische Bedingungen von Wintertagen über das Jahresmittel bis zu Sommer- und Hitzetagen bewertet werden.

Infrastruktur-Szenarien

Betrachtet werden können sowohl der klassische Regenwasserrückhalt als auch ein Infrastrukturraum mit Kabeln, Leitungen und Technik sowie entsprechende Kombinationen.

Der Bericht trägt eine **Eingabe- und eine Ergebnis-Prüfsumme mittels SHA-256**. Dadurch kann der Bericht dem berechneten Projektstand zugeordnet werden.

Grenzen

Die TIB wird ausdrücklich als:

vereinfachte technische Vorabschätzung

eingeordnet.

Sie ersetzt insbesondere keine:

- standortspezifische thermische Simulation,
- Elektroplanung,
- Abdichtungsplanung,
- Brandschutzprüfung,
- KRITIS-Prüfung,
- wasserrechtliche Bewertung.

Für die Nutzung innerhalb von JUMBO BLOCK Rescue ist zusätzlich entscheidend:

Die integrierte Temp-Seite führt selbst keine TIB-Berechnung durch.

Sie stellt weder das interne Berechnungsmodell noch eine öffentlich bedienbare Oberfläche für die thermische Bilanzierung bereit.

Die erforderliche `project.dbm` kann mit den übrigen Anwendungen der Engineering Suite vorbereitet beziehungsweise gesichert werden. Die eigentliche thermische Bilanzierung erfolgt anschließend durch JUMBO BLOCK Engineering.

Bedeutung innerhalb von JUMBO BLOCK Rescue

Temp ist innerhalb von JUMBO BLOCK Rescue deshalb **nicht als autarkes Berechnungsprogramm**, sondern als Schnittstelle zwischen der lokal möglichen Projektvorbereitung und dem internen thermischen Engineering-Prozess einzuordnen.

Die Engineering-Kette ist an dieser Stelle bewusst geteilt:

Hydro → lokale hydrologische Vorplanung

Plan → lokale geometrische Projektplanung

Temp → Informations- und Übergabeseite zur internen Thermischen Infrastruktur-Bilanz

Innerhalb der Rescue-Umgebung können damit vorhandene Projektdaten gesichert und die für eine TIB erforderlichen Grundlagen vorbereitet werden.

Die eigentliche thermische Bilanzierung bleibt dagegen dem internen JUMBO BLOCK Engineering-Werkzeug vorbehalten.

Diese Abgrenzung verhindert, dass die Verfügbarkeit der Temp-Informationsseite innerhalb von JUMBO BLOCK Rescue mit der Verfügbarkeit des internen TIB-Berechnungswerkzeugs verwechselt wird.

Weiterführende Informationen

Weitere Informationen zur Thermischen Infrastruktur-Bilanz und zum JUMBO BLOCK Engineering-Service:

https://jumboblock.org/

D.6 – Inspect

Bereich: Inspektion
Programm: JUMBO BLOCK Inspect
Baseline 20261001: 1.0.0 ff.

Zweck

JUMBO BLOCK Inspect ist das Wartungs- und Inspektionsbuch der JUMBO BLOCK Engineering Suite.

Das Programm dokumentiert den Betrieb einer Anlage über ihre gesamte Lebensdauer. Es erfasst insbesondere:

- vorhandene Anlagen und Bauteile,
- vorgesehene Prüfungen,
- Prüffristen und Fälligkeiten,
- durchgeführte Inspektionen,
- Messwerte und Prüfergebnisse,
- Fotodokumentationen,
- festgestellte Mängel,
- Maßnahmen und Mängelbeseitigungen.

Inspect verbindet damit die zuvor erzeugten Engineering-Daten mit der späteren betrieblichen Dokumentation der tatsächlich vorhandenen Anlage.

Rolle innerhalb der Engineering Suite

Inspect übernimmt die **Wartungs-, Inspektions- und Lebenszyklusdokumentation** innerhalb der Engineering Suite.

Als Ausgangsbasis kann die `project.dbm` aus der vorgelagerten Engineering-Bearbeitung eingelesen werden.

Inspect liest daraus Grunddaten aus **Hydro, Plan, Static und Temp** und kann auf Grundlage dieser Informationen Positionsvorschläge für den Anlagenbaum erzeugen.

Die Projektdatei wird dabei nur gelesen. Die Daten der vorgelagerten Engineering-Programme werden durch Inspect nicht verändert.

Nach dem Import wird die betriebliche Dokumentation in einer eigenen Datei geführt:

inspect.dbm

Damit bleiben Engineering-Projektdaten und die spätere Wartungs- und Inspektionshistorie voneinander getrennt.

Wesentliche Funktionen

Grunddaten aus der Projektplanung

Beim Einlesen einer `project.dbm` können unter anderem übernommen werden:

- Projektname,
- Ort,
- Bauherr,
- Planer,
- Modulanzahl,
- Raster,
- Anlagenabmessungen,
- Nutzvolumen,
- Platten,
- Abdichtung,
- Bemessungswerte aus Hydro und DWA-A 117,
- Verkehrslast,
- Grundwasser,
- Betonklasse,
- Temp-Infrastruktur.

Der SHA-256 der eingelesenen `project.dbm` wird nachgerechnet. Das Ergebnis wird angezeigt und zusammen mit der Quelle gespeichert.

Anlagenbaum

Inspect bildet die reale Anlage in einem hierarchischen Anlagenbaum ab.

Positionen können beliebig tief verschachtelt werden, beispielsweise:

Anlage → Schacht → Leiter

Positionen besitzen einen Typ, eine Bezeichnung, eine Anzahl und – soweit erforderlich – ein Raster.

Zusätzliche Komponenten, die nicht Bestandteil der `project.dbm` sind, können aus dem Katalog ergänzt werden. Dazu gehören beispielsweise Schächte, Einläufe, Vorbehandlung und Zusatzprodukte.

Positionen können ergänzt, ausgetauscht, als entfallen markiert oder – sofern noch keine zugehörigen Protokolle oder Mängel vorhanden sind – gelöscht werden.

Rasterinformationen ermöglichen eine genaue räumliche Zuordnung von Mängeln innerhalb der Anlage.

Plan- und Grenzwerte

Für einzelne Prüfpunkte können positionsbezogen festgelegt werden:

- Planwerte,
- Einheiten,
- Bezeichnungen,
- Mindestwerte,
- Höchstwerte.

Messwerte außerhalb definierter Grenzwerte können automatisch als **„n. i. O.“** bewertet werden.

Änderungen an Plan- und Grenzwerten werden mit Datum sowie altem und neuem Wert in der Historie dokumentiert.

Abgeschlossene Protokolle behalten dagegen die zum jeweiligen Prüfzeitpunkt geltenden Werte.

Prüfkatalog

Inspect enthält einen erweiterbaren Prüfkatalog.

Die Prüfpunkte können unter anderem folgende Prüfarten verwenden:

- Sichtprüfung,
- Funktionsprüfung,
- Reinigung/Wartung,
- Messwert,
- Auswahl,
- Freitext,
- Foto.

Zusätzlich können Intervalle, Ereignisprüfungen, Einheiten, Grenzwerte, Auswahlmöglichkeiten, Sicherheitshinweise und Quellen hinterlegt werden.

Die Intervalle des Standardkatalogs sind Vorschläge. Sie ersetzen keine Hersteller-, Betreiber- oder Regelwerksvorgaben und müssen für die jeweilige Anlage geprüft und gegebenenfalls angepasst werden.

Eigene Typen und Prüfpunkte können ergänzt werden. Kataloge lassen sich außerdem exportieren und importieren.

Fälligkeiten

Für Prüfpunkte mit festgelegtem Intervall berechnet Inspect die nächste Fälligkeit.

Dabei werden unter anderem unterschieden:

- überfällig,
- innerhalb von 30 Tagen fällig,
- erstmalig ohne Ausgangsdatum,
- in Ordnung.

Als Ausgangspunkt kann das Inbetriebnahmedatum beziehungsweise die letzte abgeschlossene Prüfung dienen.

Zusätzlich unterstützt Inspect ereignisbezogene Prüfungen und Prüfungen nach baulichen Veränderungen.

Inspektionsgründe

Inspect unterscheidet verschiedene Inspektionsanlässe.

Zum Standard gehören:

- jährliche Inspektion,
- Inspektion nach Ereignis,
- Inspektion nach baulicher Veränderung,
- Abnahme / Erstinspektion,
- Gewährleistungsinspektion,
- Inspektion nach Instandsetzung,
- Sonderinspektion.

Eigene Inspektionsgründe können ergänzt und mit zusätzlichen Pflichtangaben versehen werden.

Inspektion durchführen

Eine neue Inspektion wird zunächst als Entwurf angelegt.

Abhängig vom gewählten Inspektionsgrund werden die vorgesehenen Prüfpunkte zusammengestellt.

Für die einzelnen Prüfungen können unter anderem erfasst werden:

- **i. O.**
- **n. i. O.**
- **entfällt**
- Messwerte,

- Freitexte,
- Fotos,
- Mängel.

Bei Messwerten können die hinterlegten Plan- und Grenzwerte unmittelbar am Prüfpunkt angezeigt werden.

Fotodokumentation

Fotos können direkt in die Inspektionsdatei eingebettet werden.

Inspect verkleinert die Bilder entsprechend den gewählten Einstellungen. EXIF-Informationen wie Aufnahmezeit, GPS-Position, Höhe, Blickrichtung und Gerät können erhalten bleiben und im Bericht unter dem jeweiligen Foto erscheinen.

Für die Weitergabe eines Berichts können GPS-Koordinaten ausgeblendet werden. In der eigentlichen Inspect-Datei bleiben die Daten dabei erhalten.

Mängelmanagement

Mängel können während einer Inspektion oder direkt im Bereich Mängel erfasst werden.

Zu einem Mangel können unter anderem gehören:

- betroffene Position,
- Schweregrad,
- Beschreibung,
- Frist,
- Rasterfeld,
- Bauteil,
- Fotos.

Nach der Bearbeitung können zusätzlich Maßnahme, Verantwortlicher, Erledigungsdatum, Nachweis und weitere Fotos dokumentiert werden.

Erledigte Mängel können Bestandteil einer späteren **Inspektion nach Instandsetzung** werden.

Daten und Datenaustausch

Inspect verwendet zwei klar getrennte Datenebenen.

Eingangsdaten

project.dbm

Sie dient zur Übernahme von Projekt- und Engineering-Grunddaten.

Beim erneuten Einlesen einer geänderten Projektdatei können Änderungen erkannt und vorhandene Positionen beziehungsweise Planwerte entsprechend behandelt werden.

Positionen werden bei Änderungen nicht automatisch gelöscht. Nicht mehr zutreffende Positionen können auf Wunsch als entfallen markiert werden.

Inspektionsdaten

inspect.dbm

Eine Inspect-Datei gehört zu einer Anlage und enthält unter anderem:

- Stammdaten,
- Quelle,
- Prüfkatalog,
- Anlagenbaum,
- Inspektionen,
- Mängel,
- eingebettete Medien,
- Einstellungen.

Zusätzlich stehen unter anderem zur Verfügung:

- CSV-Export von Mängeln,
- CSV-Export von Fälligkeiten,
- Katalog-Export,
- Katalog-Import.

Datenintegrität

Inspect verwendet mehrere SHA-256-basierte Prüfmechanismen.

Datei-Hash

Die `inspect.dbm` besitzt einen SHA-256-Datei-Hash nach demselben Grundprinzip wie die `project.dbm`.

Verkettete Inspektionsprotokolle

Beim Abschluss einer Inspektion wird das Protokoll unveränderlich.

Inspect vergibt eine laufende Nummer und speichert den Hash des vorherigen abgeschlossenen Protokolls. Anschließend wird ein neuer SHA-256 über das vollständige Protokoll erzeugt.

Damit entsteht eine **verkettete Protokollhistorie**.

Fotos

Jedes eingebettete Bild wird unter seinem eigenen SHA-256 abgelegt und beim Öffnen erneut geprüft.

Die Integritätsprüfung kann damit feststellen, ob Protokolle, Bilder oder Metadaten nachträglich verändert wurden.

Sie kann jedoch nicht beweisen, dass eine ursprünglich eingegebene Information oder beispielsweise eine bereits im Foto vorhandene EXIF-Angabe inhaltlich richtig war.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Inspect läuft vollständig offline im Browser und kann direkt vom USB-Stick über `file://` verwendet werden.

Es benötigt:

- kein Benutzerkonto,
- keine Cloud,
- keinen zentralen Server.

Die Inspektionsdaten verbleiben lokal auf dem verwendeten System beziehungsweise Datenträger.

Damit ist Inspect für JUMBO BLOCK Rescue besonders relevant, weil vorhandene Anlagenunterlagen und Inspektionsinformationen auch bei Ausfall der regulären IT-Infrastruktur lokal weitergeführt werden können.

Eine vorhandene `project.dbm` kann eingelesen und als Grundlage für eine neue beziehungsweise vorhandene Anlagendokumentation verwendet werden.

Die eigentliche Wartungs- und Inspektionshistorie bleibt in der separaten `inspect.dbm` erhalten.

Datensicherung

Inspect hält zusätzlich einen lokalen Zwischenstand im Browser vor.

Dieser Zwischenstand ersetzt jedoch **nicht** das Speichern der `inspect.dbm`.

Ungespeicherte Änderungen werden in der Statusleiste angezeigt.

Da Fotos direkt in die Datei eingebettet werden, kann die Größe der `inspect.dbm` mit zunehmender Anzahl von Inspektionen deutlich wachsen. Ab einer Dateigröße von etwa **50 MB** weist Inspect darauf hin.

Für eine belastbare Dokumentation sollte die aktuelle `inspect.dbm` deshalb regelmäßig gespeichert und gesichert werden.

Berichte

Inspect stellt drei A4-Berichte bereit.

Inspektionsprotokoll

Das Inspektionsprotokoll enthält unter anderem:

- Anlage und Grunddaten,
- Inspektionsgrund,
- Umfang und Zusatzangaben,
- Ergebnisse je Position,
- Plan- und Grenzwerte,
- Fotos,
- EXIF-/GPS-Angaben,
- Mängel,
- Hash-Nachweis,
- Unterschriftsfelder.

Wartungs- und Inspektionsplan

Der Wartungs- und Inspektionsplan führt die Prüfpunkte mit:

- Plan- und Grenzwerten,
- Prüfintervallen,
- letzter Prüfung,
- nächster Prüfung

zusammen.

Mängelliste

Die Mängelliste dokumentiert offene und erledigte Mängel einschließlich vorhandener Fotos.

Über den Druckdialog des Browsers können die Berichte gedruckt oder als PDF gespeichert werden.

Grenzen und Hinweise

Inspect ist ein Werkzeug zur strukturierten Wartungs- und Inspektionsdokumentation.

Die im Standardkatalog enthaltenen Prüfintervalle sind ausdrücklich **Vorschläge**. Sie ersetzen keine verbindlichen:

- Herstellervorgaben,

- Betreibervorgaben,
- Regelwerksvorgaben.

Diese müssen für die jeweilige Anlage geprüft und angepasst werden.

Die SHA-256-Prüfungen weisen nach, ob gespeicherte Protokolle, Bilder und Metadaten seit dem jeweiligen Abschluss beziehungsweise Anhängen unverändert geblieben sind. Sie bestätigen nicht automatisch die inhaltliche Richtigkeit der ursprünglich erfassten Angaben.

Fotos im **HEIC-Format** können vom Browser unter Umständen nicht gelesen werden. Das Handbuch empfiehlt deshalb JPEG beziehungsweise einen vorherigen Export als JPEG.

Technische Einordnung in JUMBO BLOCK Rescue

Inspect erweitert die Engineering-Kette über Planung und Berechnung hinaus in den **Betrieb und Lebenszyklus der Anlage**.

Die funktionale Zuordnung lautet:

Hydro / Plan / Static / Temp → technische Projekt- und Anlagengrundlagen

Inspect → Anlagenstruktur, Prüfplanung, Inspektionen, Messwerte, Fotodokumentation, Mängel und Wartungshistorie

Damit können vorhandene Engineering-Daten nicht nur gerettet und weiterverarbeitet werden. Sie können innerhalb der Rescue-Umgebung auch als Ausgangspunkt für die betriebliche Dokumentation einer Anlage dienen.

Die getrennte `inspect.dbm`, die Integritätsprüfung und die verketteten abgeschlossenen Inspektionsprotokolle unterstützen dabei eine nachvollziehbare lokale Dokumentation auch ohne Cloud- oder Serverinfrastruktur.

Weiterführende Dokumentation

JUMBO BLOCK INSPECT – Wartungs- und Inspektionsbuch · Handbuch · Version 1.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Inspect“ auf **https://jumboblock.org/.**

D.7 – Project

Bereich: Projektmanagement
Programm: JUMBO BLOCK Project
Baseline 20261001: 1.1.2 ff.

Zweck

JUMBO BLOCK Project ist das Projektmanagement-Werkzeug der JUMBO BLOCK Engineering Suite.

Es dient zur Organisation von **Ingenieurprojekten und Machbarkeitsstudien** und strukturiert ein Projekt in drei Ebenen:

- Projekt,
- Projektabschnitte,
- Projektschritte.

Abschnitte und Schritte werden mit Verantwortlichkeiten, Terminen und Bearbeitungsstatus verbunden.

Das Programm verbindet damit die technische Projektarbeit der Engineering Suite mit einer strukturierten organisatorischen Projektführung.

Rolle innerhalb der Engineering Suite

Project übernimmt die **organisatorische Projektmanagement-Ebene**.

Während Programme wie Hydro, Plan, Budget, Static und Temp fachliche Engineering-Aufgaben abbilden, verwaltet Project den übergeordneten Projektablauf.

Dazu gehören insbesondere:

- Projekt-Stammdaten,
- Projektabschnitte und Projektschritte,
- Verantwortlichkeiten,
- ausführende Personen oder Partner,
- Status,
- Termine,
- Projektpartner,
- projektbezogene Dokumente und Originaldateien,
- Statushistorie.

Mehrere Personen können unabhängig voneinander mit Kopien eines Projekts arbeiten. Die unterschiedlichen Bearbeitungsstände können anschließend über einen **additiven Import** wieder zusammengeführt werden.

Wesentliche Funktionen

Projekt-Stammdaten

Für das Gesamtprojekt können unter anderem erfasst werden:

- Projektname,
- Auftraggeber / Kunde,
- Standort,
- Projektphase,
- Gesamtverantwortlicher,
- Gesamtausführender.

Als Projektphasen stehen zur Verfügung:

- Akquise,
- Planung,
- Genehmigung,
- Ausführung,
- Abnahme.

Projektabschnitte und Projektschritte

Ein Projekt kann in beliebig viele Projektabschnitte gegliedert werden.

Innerhalb eines Abschnitts können wiederum beliebig viele einzelne Projektschritte angelegt werden.

Damit lassen sich beispielsweise Planung, Genehmigungsverfahren und Ausführung voneinander trennen und anschließend in konkrete Arbeitsschritte untergliedern.

Abschnitte und Schritte können für eine bessere Übersicht ein- und ausgeklappt werden.

Verantwortlicher und Ausführender

Jeder Abschnitt und jeder Projektschritt besitzt zwei Rollen:

Verantwortlich
trägt die Verantwortung für den jeweiligen Eintrag.

Ausführend
führt die eigentliche Tätigkeit aus.

Beide Rollen können derselben oder unterschiedlichen Personen beziehungsweise Projektpartnern zugewiesen werden.

Fehlt eine der beiden Zuweisungen, kennzeichnet Project den Eintrag mit:

„⚠ nicht zugewiesen“

Offene Zuständigkeiten bleiben dadurch unmittelbar sichtbar.

Status und Termine

Für Abschnitte und Schritte stehen drei Status zur Verfügung:

- Offen,
- In Arbeit,
- Erledigt.

Ein Termin muss zu Beginn nicht zwingend festgelegt werden.

Soll ein Eintrag jedoch auf **„Erledigt“** gesetzt werden, ist ein Datum erforderlich.

Zusätzlich kann ein optionaler Starttermin hinterlegt werden.

Das Feld **„Reihenfolge“** ermöglicht ergänzende dokumentarische Angaben wie beispielsweise „nach Bodengutachten“. Diese Angabe steuert den Projektablauf jedoch nicht automatisch.

Termine-Übersicht

Project fasst alle Abschnitte und Schritte mit hinterlegtem Datum in einer chronologischen Terminübersicht zusammen.

Angezeigt werden:

- Datum,
- Titel,
- Typ,
- Status,
- Verantwortlicher,
- Ausführender.

Überfällige Einträge, deren Termin bereits vergangen und deren Status noch nicht „Erledigt“ ist, werden hervorgehoben.

Automatischer Zeitplan

Aus den hinterlegten Start- und Endterminen erzeugt Project automatisch eine **Gantt-Ansicht**.

Abschnitte bilden dabei die übergeordneten Zeilen, die zugehörigen Projektschritte werden darunter dargestellt.

Die Balken zeigen den Zeitraum vom Starttermin bis zum festgelegten Termin.

Der jeweilige Status wird im Zeitplan visuell dargestellt. Zusätzlich markiert eine senkrechte Linie das aktuelle Datum.

Partner und Rollen

Im Bereich **„Partner & Rollen“** können externe Projektbeteiligte erfasst werden.

Zu jedem Partner können hinterlegt werden:

- Name,
- Rolle,
- freie Notiz.

Die Partnerliste dient gleichzeitig als Referenz für die Zuordnung von Verantwortlichen und Ausführenden.

Dokumenten-Register und Dateianhänge

Project besitzt ein integriertes Dokumenten-Register.

Projektbezogene Unterlagen können mit:

- Bezeichnung,
- Notiz,
- optionaler Originaldatei

erfasst werden.

An einen Dokumenteintrag kann eine tatsächliche Datei angehängt werden, beispielsweise:

- PDF,
- Bild,
- Office-Dokument,
- andere projektbezogene Dateien.

Mehrere Dateien können gleichzeitig hinzugefügt werden. Project erzeugt dabei automatisch entsprechende Dokumenteinträge.

Die Dateien werden **direkt zusammen mit dem Projekt exportiert** und sind damit Bestandteil der `.jbpr`-Projektdatei.

Angehängte Dateien werden nach einem Import zum Download bereitgestellt. Ein direktes Öffnen im Browser wird bewusst nicht verwendet, da der Dateityp nach dem Import aus dem ZIP-Container nicht in jedem Fall zuverlässig erkannt werden kann.

Datei und Dokumenteintrag können getrennt voneinander behandelt werden: Eine angehängte Datei kann gelöst werden, ohne den Dokumenteintrag zu entfernen.

Statusprotokoll

Project führt ein fortlaufendes additives Statusprotokoll.

Dokumentiert werden unter anderem:

- Anlage von Abschnitten,
- Anlage von Projektschritten,
- Exporte,
- Importe,
- Konfliktlösungen,
- eigene manuelle Protokolleinträge.

Die Einträge enthalten Zeitstempel und Autor.

Bereits vorhandene Protokolleinträge werden nicht nachträglich verändert oder gelöscht. Dadurch entsteht eine chronologische Projektgeschichte.

Der oben im Programm eingetragene Benutzername wird als Autor für neue Protokolleinträge und weitere relevante Änderungen verwendet.

Daten und Datenaustausch

Project verwendet ein eigenes Projektdateiformat:

.jbpr

Beim Export wird das gesamte Projekt einschließlich der angehängten Originaldateien in einem ZIP-Container gespeichert.

Der Dateiname folgt dem Schema:

JJJJMMTT_HHMMSS_projektname.jbpr

Intern enthält die `.jbpr`:

- `manifest.json` mit den Projektdaten und der Prüfsumme,
- Verzeichnis `attachments/` mit den angehängten Originaldateien.

Die Datei kann beispielsweise über:

- USB-Datenträger,
- E-Mail,
- gemeinsame Dateiablage

weitergegeben werden.

Für die Verarbeitung selbst ist kein Online-Dienst erforderlich.

Additiver Import und Zusammenführen

Eine besondere Funktion von Project ist das **additive Zusammenführen unterschiedlicher Projektstände**.

Über **„Import / Zusammenführen“** kann eine `.jbpr` oder eine reine JSON-Manifestdatei eingelesen werden.

Der vorhandene lokale Projektstand wird dabei nicht einfach überschrieben.

Stattdessen werden die beiden Datenstände miteinander verglichen:

- nur in der eingehenden Datei vorhandene Einträge werden übernommen,
- identische Einträge bleiben unverändert,
- unterschiedlich bearbeitete Einträge werden als Konflikt gekennzeichnet.

Auch das Statusprotokoll wird additiv zusammengeführt. Eindeutige IDs verhindern dabei doppelte Protokolleinträge.

Angehängte Dateien aus der importierten Projektdatei werden ebenfalls übernommen.

Konfliktlösung

Werden widersprüchliche Änderungen erkannt, zeigt Project die lokale und die eingehende Version gegenübergestellt an.

Für jeden Konflikt stehen drei Möglichkeiten zur Verfügung:

Meine Version behalten

Die lokale Version bleibt aktiv.

Eingehende Version übernehmen

Die importierte Version wird zum aktiven Stand.

Beide behalten (als Duplikat prüfen)

Beide Versionen bleiben zunächst erhalten und können anschließend manuell geprüft werden.

Eine verworfene Version wird dabei nicht vollständig aus der Projekthistorie entfernt. Sie erhält den Status **„verworfen“** und bleibt mit Autor, Zeitpunkt und Begründung nachvollziehbar.

Die Entscheidung über einen Konflikt trifft die Person, die den Import durchführt. Project automatisiert bewusst keinen wechselseitigen Bestätigungsprozess zwischen Projektbeteiligten.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Project läuft vollständig lokal im Browser.

Es verwendet:

- keine Cloud,
- keinen zentralen Server,
- keine automatische Datenübertragung.

Der aktuelle Arbeitsstand wird fortlaufend im lokalen Browserspeicher `localStorage` zwischengespeichert.

Für die dauerhafte Sicherung und den Austausch zwischen Geräten oder Projektbeteiligten dient die `.jbpr`-Datei.

Diese Architektur ist für JUMBO BLOCK Rescue besonders relevant.

Nach einem Ausfall der regulären IT-Infrastruktur kann ein vorhandenes `.jbpr`-Projekt lokal geöffnet und weiterbearbeitet werden. Projektstruktur, Termine, Verantwortlichkeiten, Statusprotokoll und eingebettete Originaldateien stehen dabei gemeinsam in einer einzigen Projektdatei zur Verfügung.

Damit kann nicht nur technische Engineering-Arbeit fortgesetzt werden, sondern auch die organisatorische Projektsteuerung.

Datensicherung

Der lokale Browserspeicher ist lediglich geräte- und browserbezogen.

Für eine belastbare Datensicherung muss das Projekt deshalb regelmäßig als `.jbpr` exportiert werden.

Vor umfangreichen Zusammenführungen mit anderen Projektständen empfiehlt das Handbuch ausdrücklich einen Sicherungs-Export des aktuellen eigenen Standes.

Beim Zurücksetzen eines Projekts fragt Project zunächst, ob der aktuelle Stand exportiert werden soll. Erst anschließend werden Projektdaten, Abschnitte, Schritte, Partner, Dokumente, Anhänge und Statusprotokoll geleert.

Berichte

Project stellt zwei unterschiedliche Druckausgaben bereit.

Projektbericht

Über **„Druck / PDF“** wird ein druckfertiger Bericht erzeugt.

Dieser enthält:

- Projekt-Stammdaten,
- Abschnitte,
- Projektschritte,
- chronologische Terminliste,

- Partner,
- Dokumenten-Register,
- Statusprotokoll,
- SHA-256-Prüfsumme.

Der Bericht kann über den Browser gedruckt oder als PDF gespeichert werden.

Gantt-Bericht

Über **„Druck Gantt / PDF“** wird der Zeitplan als eigenständiges druckfertiges Dokument im Querformat erzeugt.

Das Papierformat wird abhängig von der Projektgröße automatisch zwischen **DIN A4 und DIN A0 quer** gewählt.

Damit sollen auch umfangreiche Projektpläne einschließlich ihrer Balkenbeschriftungen vollständig dargestellt werden.

Datenintegrität

Project berechnet fortlaufend eine **SHA-256-Prüfsumme über den aktuellen Projektstand**.

Die Prüfsumme wird:

- im Programm angezeigt,
- in exportierte Projektdateien aufgenommen,
- im Druckbericht ausgegeben.

Bei einer inhaltlichen Änderung verändert sich auch die Prüfsumme.

Identische Prüfsummen kennzeichnen damit identische Projektstände.

Die SHA-256-Prüfsumme ist jedoch ausdrücklich **kein Ersatz für eine Versionskontrolle**. Sie dient als Plausibilitäts- und Integritätsnachweis für den jeweiligen Datenstand.

Grenzen und Hinweise

Project besitzt keine automatische Synchronisation zwischen unterschiedlichen Geräten.

Der Austausch erfolgt bewusst über Export und Import der `.jbpr`-Dateien.

Der lokale `localStorage` ist an Browser und Gerät gebunden. Vor einem Wechsel des Geräts oder Browsers muss das Projekt deshalb exportiert werden.

Beim Zusammenführen zweier Projekte mit demselben Dateipfad eines Anhangs wird laut Handbuch lediglich die Metadatenebene auf Konflikte geprüft. Die eigentlichen Dateiinhalte werden nicht byteweise miteinander verglichen.

Unklare Konflikte zwischen Projektbeteiligten müssen deshalb organisatorisch geklärt werden.

Project ist außerdem nicht ausschließlich für große Projekte vorgesehen. Die Struktur kann ebenso für kleine Machbarkeitsstudien mit wenigen Abschnitten und Projektschritten verwendet werden.

Technische Einordnung in JUMBO BLOCK Rescue

Project ergänzt die fachlichen Engineering-Programme um eine **lokal verfügbare Projektmanagement- und Dokumentationsebene**.

Die funktionale Zuordnung lautet:

Hydro / Plan / Budget → fachliche Projektbearbeitung

Static / Temp → interne Engineering-Services mit Informations- und Übergabeseiten

Project → Projektstruktur, Verantwortlichkeiten, Termine, Partner, Dokumente und Projektverlauf

Die `.jbpr` verbindet dabei Projektdaten und angehängte Originaldokumente in einer transportablen Projektdatei.

Der additive Import ermöglicht außerdem eine Form der dezentralen Zusammenarbeit ohne zentrale Cloud-Infrastruktur: Beteiligte können getrennte Projektkopien bearbeiten und die Stände später kontrolliert zusammenführen.

Damit unterstützt Project innerhalb von JUMBO BLOCK Rescue sowohl die **Fortführung eines Projekts während eines IT-Ausfalls** als auch den kontrollierten Austausch von Projektständen über lokale oder anderweitig verfügbare Übertragungswege.

Weiterführende Dokumentation

JUMBO BLOCK Project – Benutzerhandbuch · Version 1.1.2

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Project“ auf **https://jumboblock.org/.**

D.8 – Tender

Bereich: Ausschreibung
Programm: JUMBO BLOCK Tender
Baseline 20261001: 4.0.0 ff.

Zweck

JUMBO BLOCK Tender ist das lokale Werkzeug der JUMBO BLOCK Engineering Suite zum Öffnen, Prüfen, Darstellen, Erstellen, Bearbeiten, Bepreisen und Speichern von Leistungsverzeichnissen auf Basis von **GAEB DA XML**.

Unterstützt werden die Datenphasen:

- X80,
- X81,
- X82,
- X83,
- X84.

Tender ermöglicht damit die Bearbeitung eines Leistungsverzeichnisses von der neutralen Arbeitsgrundlage über Leistungsbeschreibung, Kostenanschlag und Angebotsaufforderung bis zur Angebotsabgabe.

Rolle innerhalb der Engineering Suite

Tender übernimmt innerhalb der Engineering Suite die **Ausschreibungs- und Leistungsverzeichnis-Ebene**.

Es ergänzt die Engineering- und Projektmanagement-Werkzeuge um die strukturierte Bearbeitung von GAEB-DA-XML-Daten.

Die unterstützten Datenphasen besitzen unterschiedliche Aufgaben:

X80 – Universelle LV-Daten
Neutrale, preisfreie Arbeits- und Austauschphase.

X81 – Leistungsbeschreibung
Leistungsverzeichnis aus der Planung als Grundlage für Kostenanschlag und Ausschreibung.

X82 – Kostenanschlag
Interne Kostenermittlung mit Preisen des Planers beziehungsweise Auftraggebers.

X83 – Angebotsaufforderung
Leistungsverzeichnis für den Bieter. Interne X82-Kostenansätze werden nicht übernommen.

X84 – Angebotsabgabe
Rückgabe des Angebots mit Bieterpreisen und zulässigen Ergänzungen.

X80 bis X83 bilden dabei die gemeinsame strukturelle Bearbeitungsbasis. X84 ist als Angebotsabgabe geschützt und wird nicht als frei veränderbares Leistungsverzeichnis behandelt.

Wesentliche Funktionen

GAEB-X-Dateien öffnen

Tender kann folgende Dateien lokal öffnen:

- `.x80`,
- `.x81`,
- `.x82`,
- `.x83`,
- `.x84`,
- `.xml`.

Dateien können über die Dateiauswahl oder per Drag-and-drop geladen werden.

Beim Einlesen prüft Tender unter anderem:

- Dateiendung,
- XML-Syntax,
- Datenphase,
- Metadaten,
- LV-Struktur,
- Positionen.

Kann eine Datei nicht ordnungsgemäß verarbeitet werden, wird die geladene Arbeitskopie verworfen und die Ursache im Statusbereich angezeigt.

Neues Leistungsverzeichnis

Über **„Neues LV“** kann eine neue X80-Arbeitsdatei angelegt werden.

Dabei werden zunächst erfasst:

- Projektbezeichnung,
- LV-Bezeichnung,
- Währung,
- erster LV-Bereich,
- erste Position.

Für die erste Position können Menge, Einheit, Kurztext und optional ein Langtext angegeben werden.

Das erzeugte X80 steht anschließend unmittelbar als preisfreie Arbeitsdatei zur Verfügung.

LV-Struktur

In den Datenphasen X80 bis X83 kann die Struktur des Leistungsverzeichnisses bearbeitet werden.

LV-Bereiche können:

- hinzugefügt,
- bearbeitet,
- zulässigen übergeordneten Bereichen zugeordnet,
- innerhalb ihrer Ebene verschoben,
- gelöscht

werden.

Die Struktur basiert auf der durch `BoQBkdn` beziehungsweise `BoQLevel` definierten Hierarchie.

Benötigt ein neuer Unterbereich eine zusätzliche Hierarchieebene, kann diese ausdrücklich erweitert werden. Tender erzeugt keine stillschweigenden Zwischenebenen.

Nach Strukturänderungen werden die betroffenen OZ-/`RNoPart`-Strukturen neu aufgebaut.

Positionen

In X80 bis X83 können Positionen hinzugefügt und bearbeitet werden.

Bearbeitbar sind:

- Menge,
- Einheit,
- Kurztext,
- Langtext,
- Zuordnung zum LV-Endbereich.

Positionen können außerdem innerhalb ihres Bereichs verschoben oder gelöscht werden.

Nach Änderungen werden die Ordnungszahlen entsprechend der neuen Struktur beziehungsweise Reihenfolge neu aufgebaut.

Tender arbeitet dabei mit einer Arbeitskopie und prüft Strukturänderungen vor der Übernahme.

Leitpositionen und Unterpositionen

Tender unterstützt die GAEB-Struktur:

SumDescr / SubDescr

Eine gegliederte Position bleibt als Leitposition im Leistungsverzeichnis erhalten. Die zugehörigen Unterpositionen werden darunter geführt und über `SubDNo` nummeriert.

Unterpositionen können:

- hinzugefügt,
- bearbeitet,
- gelöscht,
- umsortiert

werden.

Eine vorhandene nachfolgende Position kann außerdem als Unterposition der unmittelbar vorhergehenden Position übernommen werden.

Unterpositionen können Menge, Einheit, Kurztext, Langtext und – in preisfähigen Arbeitsphasen – einen eigenen Einheitspreis besitzen.

Der Gesamtpreis einer Unterposition wird aus:

Menge × Einheitspreis

gebildet.

Die Leitposition erhält keinen zusätzlichen eigenen Einheitspreis. Ihr Gesamtpreis entspricht der Summe der Gesamtpreise ihrer Unterpositionen.

Suche und Navigation

Positionen können nach:

- OZ,
- Bezeichnung,
- Text

gefiltert werden.

Über die LV-Struktur kann direkt zu einem Bereich navigiert werden.

Die Tabellenansicht unterstützt 50, 100 oder 250 Positionen pro Seite.

Mit **„Ansicht zurücksetzen“** werden Such- und Strukturfilter aufgehoben, ohne die geladene Datei zu löschen.

Preise und Summen

Die Währung wird im Kopfbereich angezeigt und kann in den bearbeitbaren Arbeitsphasen geändert werden.

In X81, X82 und X83 können Einheitspreise als Arbeitswerte unmittelbar in Tender eingegeben werden.

Für normale Positionen gilt:

GP = Menge × EP

Bei Leitpositionen ergibt sich der Gesamtpreis aus den Gesamtpreisen der Unterpositionen.

Tender berechnet daraus:

- Positionssummen,
- Bereichssummen,
- LV-Gesamtsumme.

Die Bedeutung der Preise hängt von der jeweiligen Datenphase ab.

X81-Preise können insbesondere als Arbeitswerte zur Erzeugung eines X82-Kostenanschlags dienen.

X82 enthält interne Kostenansätze.

X83-Preise können als Arbeitswerte für die spätere X84-Angebotsabgabe verwendet werden.

Beim Speichern als X80, X81 oder X83 werden Preisangaben entfernt. Tender weist vor dem Export darauf hin.

Datenphasen und Phasenwechsel

Tender kann X80, X81, X82 und X83 aus den jeweils zulässigen Ausgangsphasen erzeugen.

X80

X80 bleibt ein preisfreier Arbeits- und Austauschstand.

Wird eine bepreiste Phase als X80 gespeichert, werden vorhandene Preise nach ausdrücklicher Warnung entfernt.

X82

Beim Speichern als X82 werden vorhandene Arbeits- beziehungsweise Kostenpreise einschließlich der Unterpositionspreise gespeichert.

X82 → X83

Beim Wechsel von X82 nach X83 werden die internen Kostenansätze nicht übernommen.

Damit bleiben interne Auftraggeber- beziehungsweise Planerkosten von der Angebotsaufforderung getrennt.

X84

X84 wird nicht als Ausgangsphase für X81 bis X83 verwendet. Eine vorhandene X84 kann jedoch als X80-Arbeitsgrundlage zurückgespeichert werden.

Erzeugte Dateien werden lokal mit Zeitstempel gespeichert.

X84-Angebotsabgabe

Tender kann eine X84 aus X80 oder X83 erzeugen. Eine vorhandene X84 kann außerdem erneut lokal gespeichert werden.

Für eine X84 aus X83 müssen Preise vorhanden sein. Fehlende Preise können nach Bestätigung mit **0,00** ergänzt werden.

Bei einer direkten X84-Erzeugung aus X80 weist Tender darauf hin, dass X80 preisfrei ist. Nach ausdrücklicher Bestätigung werden die Preise zunächst auf 0,00 gesetzt.

Vor dem Export werden Bieterdaten erfasst.

Pflichtfelder sind:

- Firma / Name,
- Straße und Hausnummer,
- PLZ,
- Ort.

Optional können unter anderem Land, Ansprechpartner und E-Mail angegeben werden.

Diese Angaben werden ausschließlich in die lokal erzeugte X84 geschrieben.

Leitpositionen und Unterpositionen bleiben mit ihren preisrelevanten Daten erhalten. Die eigentliche Struktur des Leistungsverzeichnisses bleibt im X84-Arbeitsmodus geschützt.

Versteckte Migrationshilfe für D81–D84

Neben den regulär sichtbaren X80- bis X84-Funktionen erkennt Tender zusätzlich ältere **GAEB-90-Dateien** mit den Endungen:

- D81,
- D82,
- D83,
- D84.

Diese Funktion ist bewusst als **Migrationshilfe** und nicht als eigener GAEB-90-Editor ausgeführt.

Die Zuordnung erfolgt phasengleich:

D81 → X81

D82 → X82

D83 → X83

D84 → X84

Die ursprüngliche D-Datei wird dabei nicht verändert. Tender erzeugt eine lokale X-Arbeitskopie.

Da DA 1990 zeilenorientiert und GAEB DA XML elementorientiert arbeitet, handelt es sich ausdrücklich nicht um eine bloße Änderung der Dateiendung. Informationen können bei der Konvertierung verändert, zusammengeführt oder nicht übertragen werden.

Tender verlangt deshalb eine anschließende fachliche Kontrolle insbesondere von:

- Positionen,
- OZ,
- Gliederung,
- Texten,
- Mengen,
- Einheiten,
- gegebenenfalls Preisen.

Nicht standardmäßig zuordenbare Datensätze werden nicht automatisch interpretiert, sondern gemeldet und nicht übernommen.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Tender läuft vollständig im Browser.

Die Projektdaten verbleiben während der Bearbeitung im Arbeitsspeicher des Browsers.

Für Projektdaten verwendet Tender:

- keine Serverkommunikation,
- keine Cookies,
- kein `LocalStorage`,
- kein `SessionStorage`.

Das Öffnen und Speichern von GAEB-Dateien sowie die PDF-Erzeugung erfolgen lokal.

Damit kann Tender innerhalb von JUMBO BLOCK Rescue eingesetzt werden, um vorhandene Leistungsverzeichnisse auch ohne verfügbare zentrale Ausschreibungs- oder Cloud-Infrastruktur zu öffnen, zu prüfen und innerhalb des unterstützten Funktionsumfangs weiterzubearbeiten.

Datensicherung

Da Tender die Projektdaten nicht dauerhaft im Browser speichert, sind die erzeugten beziehungsweise bearbeiteten Dateien selbst die maßgebliche Grundlage für die weitere Arbeit.

Der Anwender ist verantwortlich für:

- Ablage,
- Versionierung,
- Datensicherung,
- Weitergabe

der erzeugten Dateien.

Vor umfangreichen Änderungen an einer LV-Struktur sollte eine **unveränderte Ausgangsdatei** aufbewahrt werden.

Dies gilt insbesondere bei Änderungen der `BoQLevel`-Ebenen, da diese Auswirkungen auf die LV-weite OZ-Struktur besitzen.

Drucken und PDF

Tender kann vollständig lokal einen **DIN-A4-PDF-Bericht** erzeugen.

Optional können:

- Langtexte,
- Inhaltsverzeichnis

in die Ausgabe aufgenommen werden.

Der Bericht enthält abhängig vom vorhandenen Datenstand:

- Projektinformationen,
- LV-Informationen,
- Gliederung,
- Positionen,
- Unterpositionen,
- Einheitspreise,
- Gesamtpreise,
- Bereichssummen,
- LV-Gesamtsumme.

Bei X84 können zusätzlich die Bieterangaben ausgegeben werden.

Sprache und Zahlenformat

Die Benutzeroberfläche unterstützt Deutsch und Englisch und übernimmt die Spracheinstellung aus dem globalen Menü der JUMBO BLOCK Engineering Suite.

Der Sprachwechsel betrifft unter anderem:

- Beschriftungen,
- Dialoge,
- Statusmeldungen,
- Mengenformate,
- Preisformate,
- PDF-Ausgabe.

Die intern verwendeten numerischen Werte bleiben im GAEB-/XML-kompatiblen Datenformat. Die Sprachumschaltung verändert nur Darstellung und Eingabelogik.

Grenzen und Hinweise

Tender 4.0.0 arbeitet innerhalb des implementierten Umfangs von **GAEB DA XML 3.3 / 2021-05**.

Nicht jedes optionale Schemaelement ist Bestandteil der Benutzeroberfläche.

Die interne XML- und Strukturprüfung unterstützt die Plausibilitätskontrolle, ersetzt jedoch keine externe Validierung gegen das maßgebliche XSD.

Für einen formalen Datenaustausch sollten erzeugte Dateien deshalb zusätzlich gegen das maßgebliche XSD beziehungsweise mit einem geeigneten unabhängigen Prüfwerkzeug validiert werden.

Besondere Aufmerksamkeit ist außerdem beim Wechsel zwischen Datenphasen erforderlich:

X82-Kostenansätze und X83/X84-Bieterpreise sind fachlich unterschiedliche Daten.

Die entsprechenden Warnungen beim Phasenwechsel sollten deshalb nicht übersprungen werden.

Tender wird unabhängig entwickelt. Die Bezeichnung **GAEB** beschreibt ausschließlich das unterstützte Datenformat und die veröffentlichten technischen Regeln. Eine Zertifizierung, Beauftragung, Partnerschaft oder sonstige offizielle Verbindung zum GAEB wird nicht behauptet.

Technische Einordnung in JUMBO BLOCK Rescue

Tender ergänzt die Engineering Suite innerhalb von JUMBO BLOCK Rescue um ein lokal verfügbares Werkzeug für **Leistungsverzeichnisse und GAEB-Datenaustausch**.

Die funktionale Zuordnung lautet:

Engineering-Werkzeuge → technische Projektbearbeitung

Project → organisatorisches Projektmanagement und Projektdokumentation

Tender → Leistungsverzeichnis, Ausschreibung, Kostenanschlag und Angebotsdatenaustausch

Tender ermöglicht dadurch, während eines Ausfalls regulärer IT-Infrastruktur nicht nur technische Projektunterlagen weiterzubearbeiten, sondern auch vorhandene GAEB-DA-XML-Leistungsverzeichnisse lokal zu öffnen, strukturell zu bearbeiten, zu bepreisen, zwischen unterstützten Datenphasen zu überführen und als PDF zu dokumentieren.

Die vollständig lokale Verarbeitung unterstützt dabei die Datensouveränität der Rescue-Umgebung.

Weiterführende Dokumentation

JUMBO BLOCK Tender – Benutzerhandbuch · Version 4.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Tender“ auf **https://jumboblock.org/.**

D.9 – Calc

Bereich: Berechnung
Programm: JUMBO BLOCK Calc
Baseline 20261001: 2.0.0 ff.

Zweck

JUMBO BLOCK Calc ist der wissenschaftliche Web-Rechner der JUMBO BLOCK Engineering Suite.

Das Programm kombiniert mehrere allgemeine und JUMBO-BLOCK-spezifische Berechnungswerkzeuge in einer lokalen Anwendung:

- Grundrechner,
- wissenschaftliche Funktionen,
- Prozentrechnung,
- Geometrie-Rechner,
- JUMBO BLOCK Flächenbelegung,
- Einheitenumrechnung,
- JUMBO BLOCK Modulrechner,
- Berechnungsverlauf,
- Export und Import des Berechnungsverlaufs,
- PDF-Report.

Calc dient damit als schnell verfügbares Hilfswerkzeug für technische Berechnungen, ohne dafür ein externes Rechenprogramm oder einen Online-Dienst zu benötigen.

Rolle innerhalb der Engineering Suite

Calc ist ein **eigenständiges Berechnungswerkzeug** der Engineering Suite.

Anders als die projektspezifischen Engineering-Anwendungen arbeitet Calc nicht mit der gemeinsamen `project.dbm`.

Es findet kein direkter Datenaustausch mit Hydro, Plan, Budget, Static oder Temp statt.

Calc besitzt stattdessen ein eigenes Dateiformat zur Sicherung und Wiederherstellung seines Berechnungsverlaufs.

Damit ist die funktionale Trennung eindeutig:

Hydro, Plan, Budget sowie die Engineering-Services → projektspezifische Fachbearbeitung

Calc → allgemeine technische Hilfsberechnungen

Wesentliche Funktionen

Grundrechner

Der Grundrechner unterstützt:

- Addition,
- Subtraktion,
- Multiplikation,
- Division,
- Klammern,
- Dezimalzahlen,
- Vorzeichenwechsel.

`CE` löscht die aktuelle Eingabe, während `C` den gesamten Ausdruck entfernt.

Eine Rücktaste ermöglicht das zeichenweise Löschen.

Öffnende Klammern ergänzen bei Bedarf automatisch ein Multiplikationszeichen. Fehlende schließende Klammern werden beim Ausführen einer Berechnung automatisch ergänzt.

Jede mit = abgeschlossene Berechnung wird in den Berechnungsverlauf aufgenommen.

Wissenschaftliche Funktionen

Calc stellt zusätzlich wissenschaftliche Rechenfunktionen bereit:

- π,
- Quadratwurzel,
- Quadrat,
- Kubik,
- allgemeine Potenz,
- Kehrwert,
- direkte Prozentumrechnung,
- wissenschaftliche Zehnerpotenz-Eingabe,
- Sinus,
- Kosinus,
- Tangens,
- Arkussinus,

- Arkuskosinus,
- Arkustangens.

Die Winkelfunktionen arbeiten in **Grad**, nicht in Radiant.

Für ungültige Wertebereiche beziehungsweise mathematisch nicht definierte Situationen verweigert Calc die Berechnung.

Speicherfunktionen

Für Zwischenergebnisse stehen die klassischen Speicherfunktionen zur Verfügung:

- `MC` – Speicher löschen,
- `MR` – Speicher abrufen,
- `M+` – aktuellen Wert addieren,
- `M−` – aktuellen Wert subtrahieren.

Ein eingeblendetes `M` zeigt an, dass sich ein von null verschiedener Wert im Speicher befindet.

Prozentrechnung

Die Prozenttaste des Grundrechners arbeitet kontextabhängig.

Bei Addition oder Subtraktion wird ein nachfolgender Prozentwert auf die vorhergehende Basiszahl bezogen.

Beispiel:

200 + 10 % = 220

In anderen Zusammenhängen wird der eingegebene Wert durch 100 geteilt.

Die wissenschaftliche Funktion `x%` führt dagegen unabhängig vom vorhergehenden Operator immer eine direkte Division durch 100 aus.

Geometrie-Rechner

Der Geometrie-Rechner stellt fünf Berechnungsarten bereit:

- Rechteckfläche,
- Quadervolumen,
- Kreisfläche,
- Zylindervolumen,
- Dreiecksfläche.

Je nach Berechnungsart werden nur die benötigten Eingabefelder aktiviert.

Als Basiseinheiten stehen zur Verfügung:

- mm,
- cm,
- m.

Das Ergebnis wird entsprechend als Fläche oder Volumen in der gewählten Basiseinheit ausgegeben.

Alle Eingabewerte einer Berechnung müssen in derselben gewählten Einheit vorliegen. Eine automatische Umrechnung zwischen einzelnen Eingabefeldern findet nicht statt.

JUMBO BLOCK Flächenbelegung

Für Rechteck-, Kreis- und Dreiecksflächen kann Calc zusätzlich bestimmen, wie viele quadratische JUMBO BLOCK Module vollständig innerhalb einer vorgegebenen Fläche angeordnet werden können.

Als Modulgrundmaß verwendet Calc:

2,5 × 2,5 m

Zusätzlich kann ein **Arbeitsraum je Rand** angegeben werden.

Der Standardwert beträgt:

0,50 m

Dieser umlaufende Abstand wird vor der Modulbelegung von der nutzbaren Fläche abgezogen.

Bei Rechteckflächen wird die Modulanzahl direkt aus nutzbarer Länge und Breite bestimmt.

Bei Kreis- und Dreiecksflächen verwendet Calc ein rasterbasiertes Suchverfahren mit mehreren möglichen Rasterversätzen und wählt die Variante mit der höchsten Anzahl vollständig innerhalb der Fläche liegender Module.

Als Ergebnis werden ausgegeben:

- Modulanzahl,
- belegte Grundfläche,
- nicht belegte Restfläche,
- Netto-Volumen,
- Brutto-Volumen,
- Gesamtgewicht.

Die Berechnung wird ebenfalls in den Verlauf aufgenommen.

Einheitenumrechnung

Calc stellt 14 feste technische Umrechnungen bereit:

- mm ↔ m,
- cm ↔ m,
- m² ↔ ha,
- l ↔ m³,
- kg ↔ t,
- kN ↔ N,
- MPa ↔ N/mm².

Das Ergebnis wird in der jeweiligen Zieleinheit ausgegeben und automatisch in den Berechnungsverlauf übernommen.

JUMBO BLOCK Modulrechner

Der Modulrechner berechnet aus einer eingegebenen Modulanzahl unmittelbar die zugehörigen technischen Kennwerte des Standardmoduls.

Hinterlegt sind:

Netto-Volumen: 13,9 m³ je Modul

Brutto-Volumen: 15,625 m³ je Modul

Gesamtgewicht: 4,205 t je Modul

Die Ergebnisse werden bereits während der Eingabe aktualisiert.

Mit = wird der aktuelle Berechnungsstand zusätzlich in den Verlauf übernommen.

Berechnungsverlauf

Abgeschlossene Berechnungen werden fortlaufend nummeriert im Berechnungsverlauf gespeichert.

Erfasst werden Berechnungen aus:

- Grundrechner,
- Geometrie-Rechner,
- JUMBO BLOCK Flächenbelegung,
- Einheitenumrechnung,
- Modulrechner.

Neben dem ursprünglichen Ausdruck beziehungsweise den Eingaben werden die vollständigen Ergebnisse gespeichert.

Berechnungen mit mehreren Kennwerten zeigen diese als einzelne Detailzeilen.

Einzelne numerische Ergebnisse aus Grundrechnung oder Einheitenumrechnung können durch Anklicken erneut in den Grundrechner übernommen und weiterverwendet werden.

„Verlauf löschen“ entfernt den gesamten Verlauf unmittelbar und ohne zusätzliche Rückfrage.

Daten und Datenaustausch

Calc verwendet **nicht** die gemeinsame `project.dbm` der Engineering Suite.

Der Berechnungsverlauf kann stattdessen in einer eigenen Datei exportiert werden.

Das Dateinamenschema lautet:

JJJJMMTT_HHMMSS_calc.dbm

Trotz der Dateiendung handelt es sich ausdrücklich um ein **CALC-spezifisches Format** und nicht um einen Ausschnitt der gemeinsamen `project.dbm`.

Innerhalb der Datei kennzeichnet:

"application": "CALC"

das Datenformat.

Beim Import prüft Calc, ob eine gültige Calc-Datei vorliegt.

Ein erfolgreicher Import ersetzt den aktuell angezeigten Berechnungsverlauf vollständig durch den Inhalt der importierten Datei.

Für dieses Calc-Dateiformat findet laut Handbuch **keine SHA-256-Hash-Prüfung** statt.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Calc läuft vollständig lokal im Browser als einzelne HTML-Datei.

Es benötigt:

- keine Cloud,
- keinen Server,
- kein Benutzerkonto.

Damit steht innerhalb von JUMBO BLOCK Rescue auch bei Ausfall regulärer IT-Infrastruktur ein allgemeiner technischer Rechner zur Verfügung.

Das ist insbesondere für schnelle Hilfsrechnungen, Einheitenumrechnungen, geometrische Abschätzungen und JUMBO-BLOCK-bezogene Mengenabschätzungen relevant.

Calc ersetzt dabei bewusst nicht die spezialisierten Engineering-Anwendungen. Es ergänzt sie um ein unabhängiges lokales Rechenwerkzeug.

Datensicherung

Der Berechnungsverlauf wird nicht dauerhaft im Browser gespeichert.

Er befindet sich ausschließlich im Arbeitsspeicher der geöffneten Seite.

Beim:

- Neuladen,
- Schließen der Seite

geht ein nicht exportierter Verlauf verloren.

Soll ein Berechnungsverlauf erhalten bleiben, muss er deshalb bewusst über **„Export“** als Calc-Datei gespeichert werden.

Die gewählte Sprache wird davon unabhängig lokal im Browser gespeichert.

PDF-Report

Calc kann aus dem aktuellen Berechnungsverlauf einen eigenständigen druckfertigen Bericht erzeugen.

Der PDF-Report enthält:

- Datum und Uhrzeit der Erstellung,
- vollständigen nummerierten Berechnungsverlauf,
- verwendete Ausdrücke beziehungsweise Eingaben,
- sämtliche Ergebniszeilen,
- Copyright-Hinweis.

Über den Browser-Druckdialog kann der Bericht gedruckt oder als PDF gespeichert werden.

Ist der Berechnungsverlauf leer, weist der Bericht entsprechend darauf hin.

Sprache und Bedienung

Calc unterstützt Deutsch und Englisch.

Die Sprachumschaltung erfolgt über das globale Menü der Engineering Suite.

Dabei werden unter anderem angepasst:

- Beschriftungen,
- Dezimaltrennzeichen,
- PDF-Report.

Der vorhandene Berechnungsverlauf wird durch einen Sprachwechsel nicht verändert.

Neben den Bildschirmtasten unterstützt Calc auch eine weitgehende Tastaturbedienung mit Ziffern, Rechenoperatoren, Klammern, Prozenttaste, Enter, Rücktaste, Escape und Entf.

Grenzen und Hinweise

Calc ist ein Hilfsrechner und kein Ersatz für die projektspezifischen Engineering-Programme.

Insbesondere findet kein Datenaustausch mit der gemeinsamen `project.dbm` statt.

Die JUMBO BLOCK Flächenbelegung für Kreis- und Dreiecksflächen verwendet ein **rasterbasiertes Näherungsverfahren** mit einer begrenzten Anzahl getesteter Rasterversätze.

Bei sehr großen Flächen wird die Anzahl der getesteten Positionen aus Rechenzeitgründen reduziert.

Das Ergebnis stellt damit eine Näherung für eine axial ausgerichtete Modulbelegung dar und keine Optimierung über beliebige Rotationswinkel.

Da der Berechnungsverlauf nicht dauerhaft gespeichert wird, sollte er vor dem Schließen oder Neuladen der Anwendung exportiert werden, wenn er später noch benötigt wird.

Das Löschen des Verlaufs erfolgt unmittelbar und ohne Rückfrage.

Versionshinweis der Baseline

Das Handbuch weist für den eingefrorenen Programmstand auf eine Versionsabweichung hin:

Die Programmoberfläche trägt **Version 2.0.0**. In der exportierten Calc-Datei und im PDF-Report wird im zugrunde liegenden Programmstand dagegen **Version 1.7.7** ausgegeben.

Diese Diskrepanz besteht laut Handbuch im aktuellen Quellcode und ist beim Abgleich von Versionsangaben zu berücksichtigen.

Für das JUMBO BLOCK Rescue Handbuch gilt entsprechend der eingefrorenen Baseline:

JUMBO BLOCK Calc 2.0.0 ff.

Technische Einordnung in JUMBO BLOCK Rescue

Calc stellt innerhalb von JUMBO BLOCK Rescue einen **allgemeinen lokalen technischen Rechner** bereit.

Die funktionale Abgrenzung lautet:

Hydro → hydrologische Vorbemessung

Plan → geometrische Anlagenplanung

Budget → Kostenabschätzung

Static / Temp → interne Engineering-Services

Calc → allgemeine Berechnungen, Geometrie, Einheiten und schnelle JUMBO BLOCK Mengenabschätzungen

Calc kann damit unabhängig von einer vorhandenen Projektdatei eingesetzt werden und eignet sich besonders für spontane Kontroll-, Plausibilitäts- und Hilfsrechnungen während der lokalen Arbeit mit JUMBO BLOCK Rescue.

Weiterführende Dokumentation

JUMBO BLOCK Calc – Benutzerhandbuch · Version 2.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Calc“ auf **https://jumboblock.org/.**

D.10 – Units

Bereich: Einheiten
Programm: JUMBO BLOCK Units
Baseline 20261001: 2.0.0 ff.

Zweck

JUMBO BLOCK Units ist der umfassende Einheitenumrechner der JUMBO BLOCK Engineering Suite für Engineering und Physik.

Das Programm umfasst **26 Kategorien in vier Gruppen**:

- Mechanik & Physik,
- Elektrotechnik,
- Thermodynamik,
- Sonstiges.

Im Unterschied zu klassischen Umrechnern mit jeweils einer Quell- und Zieleinheit wird ein Wert in einer Ausgangseinheit eingegeben. Units zeigt anschließend **sofort alle Einheiten der gewählten Kategorie gleichzeitig umgerechnet an**.

Damit können technische Einheiten schnell verglichen und umgerechnet werden, ohne einzelne Zielgrößen nacheinander auswählen zu müssen.

Rolle innerhalb der Engineering Suite

Units ist ein **eigenständiges Hilfswerkzeug** der Engineering Suite.

Es tauscht keine Daten mit der gemeinsamen `project.dbm` der projektspezifischen Suite-Werkzeuge aus.

Der Berechnungsverlauf wird stattdessen in einem eigenen, Units-spezifischen Dateiformat exportiert und importiert.

Die funktionale Abgrenzung lautet damit:

Hydro, Plan, Budget sowie die Engineering-Services → projektspezifische Fachbearbeitung

Calc → allgemeine technische Berechnungen

Units → physikalische und technische Einheitenumrechnung

Wesentliche Funktionen

Kategorien

Die 26 Kategorien sind in vier Gruppen gegliedert.

Mechanik & Physik

Diese Gruppe umfasst 18 Kategorien:

- Länge,
- Fläche,
- Volumen,
- Masse,
- Kraft,
- Druck / Spannung,
- Energie / Arbeit,
- Leistung,
- Drehmoment,
- Geschwindigkeit,
- Beschleunigung,
- Dichte,
- Winkel,
- Zeit,
- Volumenstrom,
- Frequenz / Drehzahl,
- Dynamische Viskosität,
- Kinematische Viskosität.

Elektrotechnik

Die elektrotechnischen Kategorien sind:

- Elektrische Spannung,
- Elektrische Stromstärke,
- Elektrischer Widerstand,
- Elektrische Ladung.

Thermodynamik

Die Gruppe Thermodynamik enthält:

- Temperatur.

Sonstiges

Zusätzlich stehen zur Verfügung:

- Datenmenge,
- Beleuchtungsstärke,
- Kraftstoffverbrauch.

Die einzelnen Kategorien enthalten je nach physikalischer Größe zwischen 3 und 18 Einheiten.

Umrechnung durchführen

Die Umrechnung wird durch drei Eingaben bestimmt:

- Kategorie,
- Zahlenwert,
- Ausgangseinheit „Von“.

Jede Änderung löst automatisch eine Neuberechnung aus. Ein gesonderter Berechnen-Knopf ist nicht erforderlich.

Die Ergebnisliste zeigt für jede Einheit der gewählten Kategorie:

- Einheitensymbol,
- ausgeschriebenen Namen,
- umgerechneten Wert.

Die gewählte Ausgangseinheit wird in der Ergebnisliste hervorgehoben.

Beim Wechsel einer Kategorie wird automatisch die erste passende Einheit als neue Ausgangseinheit gewählt und die Ergebnisliste neu berechnet.

Zahlenformatierung

Units passt die Darstellung der Ergebnisse automatisch an die jeweilige Größenordnung an.

Sehr kleine beziehungsweise sehr große Werte werden bei Bedarf in wissenschaftlicher Schreibweise dargestellt.

Je nach Größenordnung verwendet das Programm unterschiedliche Anzahlen von Nachkommastellen.

Die Darstellung erfolgt außerdem sprachabhängig:

Deutsch → Dezimalkomma

Englisch → Dezimalpunkt

Die Formatierung betrifft die Darstellung und verändert nicht die zugrunde liegenden Umrechnungswerte.

Sonderfall Temperatur

Temperaturwerte können nicht ausschließlich über einen einfachen linearen Faktor ineinander umgerechnet werden, da die Temperaturskalen unterschiedliche Nullpunkte besitzen.

Units verwendet deshalb eigene Umrechnungsformeln für:

- Celsius,
- Fahrenheit,
- Kelvin,
- Rankine.

Damit werden neben den Skalierungsunterschieden auch die jeweiligen Nullpunktverschiebungen berücksichtigt.

Sonderfall Kraftstoffverbrauch

Auch der Kraftstoffverbrauch unterscheidet sich von einer gewöhnlichen linearen Einheitenumrechnung.

Unterstützt werden:

- Liter/100 km,
- Kilometer/Liter,
- mpg (US),
- mpg (UK).

Diese Größen stehen teilweise in einem **reziproken Verhältnis** zueinander.

Units verwendet deshalb eigene Umrechnungsformeln.

Bei einem Verbrauch von **0 l/100 km** ist die Umrechnung in die kehrwertigen Einheiten mathematisch nicht definiert und wird als Fehler dargestellt.

Berechnungsverlauf

Die aktuell angezeigte Umrechnung kann mit:

„In Verlauf speichern“

bewusst in den Berechnungsverlauf übernommen werden.

Anders als bei Calc geschieht dies **nicht automatisch** bei jeder Berechnung.

Ein Verlaufseintrag enthält:

- Kategorie,
- eingegebenen Wert,

- Ausgangseinheit,
- sämtliche umgerechneten Einheiten.

Ein vorhandener Verlaufseintrag kann angeklickt werden. Units stellt daraufhin die zugehörige Kategorie, Ausgangseinheit und den ursprünglichen Wert wieder im Hauptformular her und berechnet die Ergebnisse erneut.

„Verlauf löschen“ entfernt den gesamten Berechnungsverlauf unmittelbar und ohne zusätzliche Rückfrage.

Daten und Datenaustausch

Units verwendet nicht die gemeinsame `project.dbm` der Engineering Suite.

Der Berechnungsverlauf kann stattdessen in einer eigenen Datei gespeichert werden.

Das Dateinamenschema lautet:

JJJJMMTT_HHMMSS_units.dbm

Die Datei ist ein eigenständiges **UNITS-Format**.

Innerhalb der Datei kennzeichnet:

"application": "UNITS"

den Dateityp.

Für dieses Format findet laut Handbuch **keine SHA-256-Hash-Prüfung** statt.

Import

Eine zuvor exportierte Units-Datei kann wieder importiert werden.

Beim Import prüft Units jeden Verlaufseintrag einzeln.

Dabei müssen insbesondere:

- die Kategorie in der aktuellen Programmversion vorhanden sein,
- die Ausgangseinheit existieren,
- der eingegebene Wert eine gültige Zahl sein,
- mindestens ein umgerechnetes Ergebnis numerisch gültig sein.

Ungültige einzelne Einträge werden beim Import übersprungen.

Handelt es sich insgesamt nicht um eine gültige Units-Datei, wird nichts übernommen.

Ein erfolgreicher Import **ersetzt den aktuell angezeigten Berechnungsverlauf vollständig** durch den gültigen Inhalt der importierten Datei.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Units läuft vollständig lokal im Browser als einzelne HTML-Datei.

Es benötigt:

- keine Cloud,
- keinen Server,
- kein Benutzerkonto,
- keine Online-Verbindung für die Umrechnungsfaktoren.

Die verwendeten physikalischen Umrechnungsfaktoren sind fest im Werkzeug hinterlegt.

Damit steht innerhalb von JUMBO BLOCK Rescue ein unabhängiger technischer Einheitenumrechner zur Verfügung, selbst wenn reguläre Engineering-Arbeitsplätze oder Online-Werkzeuge nicht erreichbar sind.

Gerade bei Diagnose-, Planungs- und technischen Rescue-Arbeiten können damit Werte aus unterschiedlichen Einheitensystemen unmittelbar miteinander verglichen werden.

Datensicherung

Der Berechnungsverlauf wird nicht dauerhaft im Browser zwischengespeichert.

Er existiert ausschließlich im Arbeitsspeicher der geöffneten Seite.

Beim:

- Neuladen,
- Schließen der Seite

geht ein nicht exportierter Verlauf verloren.

Soll der Verlauf erhalten bleiben, muss er deshalb bewusst über **„Export“** als Units-Datei gespeichert werden.

Die gewählte Sprache wird davon unabhängig lokal im Browser gespeichert.

Da **„Verlauf löschen“** ohne Rückfrage wirkt, sollte ein benötigter Verlauf zuvor exportiert werden.

PDF-Report

Units kann aus dem aktuellen Berechnungsverlauf einen eigenständigen druckfertigen Bericht erzeugen.

Der PDF-Report enthält:

- Datum und Uhrzeit der Erstellung,
- vollständigen nummerierten Berechnungsverlauf,
- Kategorie,
- Ausgangswert,
- sämtliche umgerechneten Einheiten,

- Copyright-Hinweis.

Innerhalb des Berichts kann über **„Drucken / PDF“** der Browser-Druckdialog geöffnet werden.

Ist der Berechnungsverlauf leer, enthält der Bericht anstelle der Berechnungsliste einen entsprechenden Hinweis.

Sprache

Units unterstützt Deutsch und Englisch.

Die Sprachumschaltung erfolgt über das globale Menü der Engineering Suite.

Dabei werden unter anderem angepasst:

- Beschriftungen,
- Kategorienamen,
- Einheitennamen,
- Zahlenformatierung,
- PDF-Report.

Der vorhandene Berechnungsverlauf bleibt beim Sprachwechsel erhalten.

Die zuletzt gewählte Sprache wird lokal im Browser gespeichert und beim nächsten Öffnen wieder verwendet.

Grenzen und Hinweise

Units ist ein Werkzeug zur **Einheitenumrechnung** und kein Ersatz für projektspezifische Engineering-Berechnungen.

Es findet kein Datenaustausch mit der gemeinsamen `project.dbm` statt.

Die Umrechnungsfaktoren sind fest im Programm hinterlegt. Externe Live-Daten werden nicht benötigt.

Für Temperatur und Kraftstoffverbrauch gelten aufgrund ihrer mathematischen Eigenschaften besondere Umrechnungsformeln.

Ein Berechnungsergebnis wird nur dann in den Verlauf übernommen, wenn ausdrücklich **„In Verlauf speichern“** gewählt wird. Das bloße Anzeigen einer Umrechnung erzeugt keinen Verlaufseintrag.

Da keine dauerhafte Speicherung des Berechnungsverlaufs im Browser erfolgt, muss ein benötigter Arbeitsstand vor dem Schließen oder Neuladen exportiert werden.

Technische Einordnung in JUMBO BLOCK Rescue

Units ergänzt JUMBO BLOCK Rescue um einen **vollständig lokalen technischen Einheitenumrechner**.

Die funktionale Zuordnung lautet:

Calc → allgemeine technische Berechnungen

Units → Einheitenumrechnung für Engineering und Physik

Die 26 Kategorien decken dabei mechanische, physikalische, elektrotechnische und thermodynamische Größen sowie weitere technische Einheiten ab.

Das Prinzip, alle Einheiten einer Kategorie gleichzeitig darzustellen, ermöglicht insbesondere bei unbekannten oder unterschiedlich dokumentierten Einheitensystemen einen schnellen Überblick.

Damit steht auch während eines IT-Ausfalls oder in einer isolierten Rescue-Umgebung ein allgemeines Umrechnungswerkzeug zur Verfügung, das vollständig ohne externe Dienste arbeitet.

Weiterführende Dokumentation

JUMBO BLOCK Units – Benutzerhandbuch · Version 2.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Units“ auf **https://jumboblock.org/.**

D.11 – Note

Bereich: Notizen
Programm: JUMBO BLOCK Note
Baseline 20261001: 1.0.0 ff.

Zweck

JUMBO BLOCK Note ist das schnelle lokale Notizprogramm der JUMBO BLOCK Engineering Suite.

Es dient zur freien Dokumentation von:

- Gedanken,
- Zwischenständen,
- Arbeitshinweisen,
- technischen Notizen,
- Verweisen auf andere Programme der Engineering Suite,
- externen Ressourcen,
- zugehörigen Dateien.

Note ist dabei nicht an ein bestimmtes technisches Fachgebiet gebunden und kann als allgemeines Notiz- und Dokumentationswerkzeug eingesetzt werden.

Rolle innerhalb der Engineering Suite

Note ergänzt die fachlichen Engineering-Programme um eine **freie lokale Dokumentationsebene**.

Während Hydro, Plan, Budget und weitere Anwendungen strukturierte Fachaufgaben bearbeiten, können in Note zusätzliche Informationen unabhängig von diesen Datenmodellen festgehalten werden.

Note tauscht keine Daten mit der gemeinsamen `project.dbm` aus.

Verweise auf andere Suite-Werkzeuge – beispielsweise `plan.html` – sind reine Links. Sie erzeugen keine technische Datenverbindung und führen keinen Datenimport durch.

Für Sicherung und Austausch verwendet Note ein eigenes Dateiformat.

Wesentliche Funktionen

Notizen erstellen und bearbeiten

Beim erstmaligen Öffnen wird automatisch eine leere Notiz angelegt.

Jede Notiz besteht aus:

- Titel,
- Notiztext,
- Links und Dateien.

Der Titel kann bis zu **200 Zeichen** enthalten.

Neue Notizen werden oben in die Notizenliste eingefügt.

Die Reihenfolge orientiert sich an der Erstellung und wird durch spätere Bearbeitungen nicht verändert.

Notizenliste

Die Notizenliste zeigt:

- Titel,
- Datum und Uhrzeit der letzten Änderung.

Wurde kein Titel vergeben, erscheint die Bezeichnung:

„Unbenannte Notiz“

Die aktuell geöffnete Notiz wird optisch hervorgehoben.

Suche

Das Suchfeld filtert die Notizenliste unmittelbar.

Durchsucht werden gemeinsam:

- Titel,
- Notiztext.

Die Suche unterscheidet nicht zwischen Groß- und Kleinschreibung.

Damit können auch größere lokale Notizsammlungen schnell nach Begriffen durchsucht werden.

Speichern

Änderungen an Titel, Text sowie Links und Dateien werden zunächst als ungespeicherte Änderungen behandelt.

Über **„Speichern“** wird die aktuelle Notiz dauerhaft in den lokalen Browserspeicher übernommen.

Dabei wird auch der Änderungszeitpunkt aktualisiert.

Zusätzlich kann die aktuelle Notiz mit:

Strg + S

beziehungsweise unter macOS mit:

Cmd + S

gespeichert werden.

Automatisches Speichern

Note speichert ungesicherte Änderungen zusätzlich automatisch:

- beim Wechsel zu einer anderen Notiz,
- beim Anlegen einer neuen Notiz,
- beim Verlassen beziehungsweise Schließen der Seite.

Dadurch werden Änderungen im normalen Arbeitsablauf weitgehend automatisch gesichert.

Bei außergewöhnlichen Situationen wie einem Browserabsturz können noch nicht gespeicherte Änderungen dennoch verloren gehen.

Links und Dateien

Jede Notiz besitzt einen eigenen Bereich **„Links und Dateien“**.

Dabei werden zwei Arten von Verknüpfungen unterstützt.

Links

Es können externe oder relative Links hinterlegt werden.

Beispiele sind:

- Web-Adressen,
- relative Pfade,
- Verweise auf andere Programme der Engineering Suite.

Für einen Link können eine Bezeichnung und das eigentliche Ziel angegeben werden.

Externe Links werden beim Aufruf in einem neuen Browser-Tab geöffnet.

Dateien

Lokale Dateien können direkt mit einer Notiz verknüpft werden.

Die ausgewählte Datei wird dabei **vollständig in die Notiz eingebettet**.

Technisch erfolgt dies als Base64-kodierte Daten-URL.

Dadurch bleibt die Datei auch dann Bestandteil der Notiz, wenn die ursprüngliche Quelldatei später nicht mehr verfügbar ist.

Eingebettete Dateien bleiben ebenfalls beim Export und anschließenden Import erhalten.

Die maximale Größe beträgt:

10 MB je Datei

Größere Dateien werden abgelehnt.

Eine eingebettete Datei kann später wieder unter ihrem ursprünglichen Dateinamen heruntergeladen werden.

Links und Dateien können einzeln aus einer Notiz entfernt werden. Die Änderung wird mit dem nächsten Speichern dauerhaft übernommen.

Daten und Datenaustausch

Note verwendet nicht die gemeinsame `project.dbm` der Engineering Suite.

Für Sicherung und Datenaustausch steht ein eigenes NOTE-Format zur Verfügung.

Der Dateiname folgt dem Schema:

JJJJMMTT_HHMMSS_note.dbm

Beim Export werden **sämtliche vorhandenen Notizen** gemeinsam gespeichert, nicht nur die aktuell geöffnete Notiz.

Die Datei enthält ein eigenes Schema:

"schema": "JUMBO_BLOCK_NOTE"

Auch eingebettete Dateien werden vollständig in diese Exportdatei übernommen.

Für das NOTE-Dateiformat findet laut Handbuch **keine SHA-256-Hash-Prüfung** statt.

Import

Eine zuvor mit Note exportierte Datei kann wieder importiert werden.

Das Programm prüft dabei, ob eine gültige NOTE-Datei vorliegt.

Fehlt das erforderliche Schema oder enthält die Datei kein gültiges Notizen-Array, wird der Import abgebrochen.

Wichtig ist das Importverhalten:

Ein erfolgreicher Import ersetzt die gesamte bisher im Browser gespeicherte Notizsammlung.

Es werden nicht lediglich einzelne Notizen ergänzt.

Der Vorgang kann nicht rückgängig gemacht werden.

Vor einem Import sollte deshalb ein Export des bestehenden Standes angelegt werden, wenn die vorhandenen Notizen erhalten bleiben sollen.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Note läuft vollständig lokal im Browser als einzelne HTML-Datei.

Es benötigt:

- keine Cloud,
- keinen Server,

- kein Benutzerkonto.

Eine Besonderheit gegenüber mehreren anderen eigenständigen Werkzeugen der Engineering Suite ist die dauerhafte lokale Speicherung der Notizen im Browser über:

localStorage

Notizen bleiben dadurch auch nach dem Schließen der Seite oder einem Neustart des Browsers erhalten, sofern derselbe Browser auf demselben Gerät verwendet wird und dessen lokale Daten nicht gelöscht werden.

Für JUMBO BLOCK Rescue ist dies besonders nützlich, weil während einer Störung oder Wiederherstellung unmittelbar lokale Arbeitsnotizen geführt werden können.

Beispielsweise lassen sich:

- Bearbeitungsstände,
- offene Aufgaben,
- technische Beobachtungen,
- Dateipfade,
- Links,
- ergänzende Dokumente

zusammen mit den jeweiligen Notizen lokal festhalten.

Datensicherung

Die Speicherung im `localStorage` ist dauerhaft, aber **browser- und gerätegebunden**.

Ein Wechsel des:

- Browsers,
- Geräts

übernimmt die vorhandenen Notizen nicht automatisch.

Auch das Löschen von Browserdaten kann die gespeicherten Notizen unzugänglich machen.

Private beziehungsweise Inkognito-Browserfenster sind ebenfalls nicht als dauerhafter Speicher geeignet.

Für eine belastbare Sicherung müssen die Notizen deshalb regelmäßig als:

***_note.dbm**

exportiert werden.

Der Export ist zugleich der vorgesehene Weg, Notizen zwischen unterschiedlichen Geräten oder Browsern auszutauschen.

Eine automatische Synchronisation findet nicht statt.

Drucken und PDF

Über **„Drucken“** wird die aktuelle Notiz zunächst automatisch gespeichert.

Anschließend erzeugt Note einen eigenständigen druckfertigen Bericht.

Dieser enthält:

- Titel,
- vollständigen Notiztext,
- verknüpfte Links,
- eingebettete Dateien mit Bezeichnung beziehungsweise Dateiname,
- Erstellungsdatum,
- Datum der letzten Änderung,
- Druckdatum und Druckuhrzeit.

Über **„Drucken / PDF“** wird anschließend der Browser-Druckdialog geöffnet.

Damit kann die Notiz ausgedruckt oder als PDF gespeichert werden.

Sprache

Note unterstützt Deutsch und Englisch.

Die Sprachumschaltung erfolgt über das globale Menü der Engineering Suite.

Dabei werden:

- Beschriftungen,
- Platzhaltertexte,
- Druckbericht

umgeschaltet.

Die gespeicherten Inhalte der Notizen selbst werden dadurch nicht verändert.

Eine Besonderheit von Note ist die tabübergreifende Sprachumschaltung:

Wird die Sprache in einem anderen gleichzeitig geöffneten Browser-Tab desselben Ursprungs geändert, übernimmt Note diese Änderung automatisch, ohne dass die Seite neu geladen werden muss.

Grenzen und Hinweise

Note ist ein freies lokales Notizwerkzeug und kein Bestandteil des gemeinsamen `project.dbm`-Datenmodells.

Links auf andere Programme der Engineering Suite sind ausschließlich Verweise und übertragen keine Daten.

Die dauerhafte Speicherung über `localStorage` ist an den jeweiligen Browser und das jeweilige Gerät gebunden. Sie ersetzt deshalb keine externe Datensicherung.

Ein Import ersetzt immer die **gesamte bestehende Notizsammlung**. Vor dem Import eines anderen Datenstandes sollte daher eine Sicherung exportiert werden.

Da eingebettete Dateien vollständig in den Notizen und damit auch in der Exportdatei gespeichert werden, können viele beziehungsweise größere Anhänge die Größe der `note.dbm` deutlich erhöhen.

Die Größenbegrenzung beträgt 10 MB pro eingebetteter Datei.

Die Notizenliste wird nach Erstellungsreihenfolge und nicht nach dem Zeitpunkt der letzten Bearbeitung sortiert. Für ältere, später erneut bearbeitete Notizen kann deshalb die integrierte Suche hilfreich sein.

Technische Einordnung in JUMBO BLOCK Rescue

Note stellt innerhalb von JUMBO BLOCK Rescue ein **allgemeines lokales Notiz- und Dokumentationswerkzeug** bereit.

Die funktionale Zuordnung lautet:

Project → strukturiertes Projektmanagement und Projektdokumentation

Note → freie Notizen, Zwischenstände, Verweise und ergänzende Dateien

Gerade während eines Rescue-Einsatzes können damit Informationen dokumentiert werden, die nicht unmittelbar in die strukturierten Datenmodelle der übrigen Engineering-Anwendungen gehören.

Durch die Kombination aus lokalem `localStorage`, manuellem Export und eingebetteten Dateien können Notizen unabhängig von Cloud- und Serverdiensten geführt und bei Bedarf als eigenständige `.dbm`-Datei gesichert beziehungsweise übertragen werden.

Weiterführende Dokumentation

JUMBO BLOCK Note – Benutzerhandbuch · Version 1.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Note“ auf **https://jumboblock.org/.**

D.12 – Tables

Bereich: Tabellen
Programm: JUMBO BLOCK Tables
Baseline 20261001: 2.0.0 ff.

Zweck

JUMBO BLOCK Tables ist die vollständig lokale Tabellenkalkulation der JUMBO BLOCK Engineering Suite.

Das Programm stellt eine eigenständige Arbeitsmappe mit mehreren Arbeitsblättern bereit und unterstützt unter anderem:

- Tabellen und Listen,
- Formeln und Berechnungen,
- Bezüge zwischen Arbeitsblättern,
- Zellformatierungen,
- bedingte Formatierungen,
- Sortieren sowie Suchen und Ersetzen,
- Balken-, Linien- und Kreisdiagramme,
- Import und Export von XLSX,
- Import und Export von CSV,
- das native `.jbtab`-Format,
- SHA-256-Integritätsprüfung,
- Druck- und PDF-Berichte.

Tables ist damit nicht auf ein bestimmtes technisches Fachgebiet beschränkt, sondern als allgemeines Tabellen- und Auswertungswerkzeug ausgelegt.

Rolle innerhalb der Engineering Suite

Tables ergänzt die spezialisierten Engineering-Anwendungen um eine **universelle lokale Tabellenkalkulation**.

Es kann beispielsweise für:

- freie Berechnungen,
- technische Tabellen,
- Mengenlisten,
- Auswertungen,

- Übersichten,
- Dokumentationslisten,
- Diagramme

verwendet werden.

Tables tauscht keine Daten mit der gemeinsamen `project.dbm` von Hydro, Plan, Budget, Static und Temp aus.

Für den eigenen Datenaustausch verwendet das Programm stattdessen:

- `.jbtab`,
- XLSX,
- CSV.

Damit bleibt Tables funktional unabhängig von den projektspezifischen Datenmodellen der Engineering Suite.

Arbeitsmappe und Arbeitsblätter

Beim ersten Start stellt Tables eine Arbeitsmappe mit drei leeren Arbeitsblättern bereit:

- Tabelle1,
- Tabelle2,
- Tabelle3.

Jedes Blatt besitzt zunächst:

- 26 Spalten von A bis Z,
- 100 Zeilen.

Weitere Arbeitsblätter können angelegt, umbenannt, gelöscht und per Ziehen umsortiert werden.

Jedes Arbeitsblatt verwaltet eigene:

- Zellen,
- Formatierungen,
- Spaltenbreiten,
- Zeilenhöhen,
- Diagramme,
- Regeln für bedingte Formatierung,
- Fixierungen.

Formeln können gleichzeitig auf Zellen anderer Arbeitsblätter derselben Arbeitsmappe zugreifen.

Wird ein Arbeitsblatt umbenannt, passt Tables vorhandene Formelbezüge auf dieses Blatt automatisch an.

Formeln und Berechnungen

Formeln beginnen mit:

=

Unterstützt werden unter anderem die Operatoren:

- Addition,
- Subtraktion,
- Multiplikation,
- Division,
- Potenz,
- Prozent,
- Textverkettung,
- Vergleichsoperatoren.

Zellbezüge und Zellbereiche können direkt in Formeln verwendet werden.

Relative und absolute Bezüge werden unterstützt, beispielsweise:

A1

A1

A$1

$A1

Beim Kopieren und Ausfüllen werden relative Zellbezüge entsprechend ihrer neuen Position angepasst.

Tables erkennt außerdem **Zirkelbezüge**, auch wenn diese über mehrere Arbeitsblätter hinweg entstehen.

Formelfunktionen

Tables besitzt einen eigenen Formelmotor mit **34 Funktionen**.

Dazu gehören unter anderem:

- SUMME / SUM,
- MITTELWERT / AVERAGE,
- MIN,
- MAX,

- ANZAHL / COUNT,
- WENN / IF,
- UND / AND,
- ODER / OR,
- RUNDEN / ROUND,
- SVERWEIS / VLOOKUP,
- INDEX,
- VERGLEICH / MATCH,
- SUMMEWENN / SUMIF,
- ZÄHLENWENN / COUNTIF,
- WENNFEHLER / IFERROR,
- ABS,
- WURZEL / SQRT,
- POTENZ / POWER,
- PI,
- LN,
- LOG,
- EXP,
- SIN,
- COS,
- TAN,
- ARCSIN / ASIN,
- ARCCOS / ACOS,
- ARCTAN / ATAN,
- BOGENMASS / RADIANS,
- GRAD / DEGREES,
- HEUTE / TODAY.

Die Funktionen können unabhängig von der gewählten Oberflächensprache mit ihren deutschen oder englischen Namen verwendet werden.

Die Winkelfunktionen arbeiten im **Bogenmaß**. Für Berechnungen mit Grad stehen entsprechende Umrechnungsfunktionen zur Verfügung.

Fehlererkennung

Tables verwendet sieben definierte Fehlerwerte:

- `#WERT!`
- `#DIV/0!`
- `#NAME?`
- `#BEZUG!`
- `#ZIRKEL!`
- `#NV`
- `#ZAHL!`

Damit werden unter anderem falsche Werttypen, Division durch null, ungültige Zellbezüge, unbekannte Funktionen, Zirkelbezüge und mathematisch ungültige Operationen gekennzeichnet.

Bearbeitung und Zwischenablage

Zellen und Zellbereiche können:

- ausgewählt,
- bearbeitet,
- kopiert,
- ausgeschnitten,
- eingefügt,
- geleert

werden.

Beim Kopieren werden Werte, Formeln und Formatierungen übernommen.

Die kopierten Zellwerte stehen zusätzlich als tabulatorgetrennter Text in der Zwischenablage des Systems zur Verfügung und können dadurch auch in andere Programme eingefügt werden.

Umgekehrt kann tabellarischer Text aus anderen Anwendungen als Werte in Tables übernommen werden.

Mit den Funktionen:

- Ausfüllen nach unten,
- Ausfüllen nach rechts

können Inhalte und Formeln einer Ausgangszeile beziehungsweise Ausgangsspalte auf einen größeren Bereich übertragen werden.

Relative Zellbezüge wandern dabei entsprechend mit.

Rückgängig und Wiederholen

Tables unterstützt bis zu **100 Schritte** für Rückgängig und Wiederholen.

Dies gilt unter anderem für:

- Zellinhalte,
- Formatierungen,
- Einfügen,
- Ausfüllen,
- Sortieren,
- Ersetzen,
- Zeilen und Spalten,
- Spaltenbreiten und Zeilenhöhen,
- verbundene Zellen,
- bedingte Formatierungen,
- Fixierungen,
- Diagramme,
- Arbeitsblätter,
- Importe,
- Zurücksetzen.

Der Rückgängig-Verlauf existiert nur im Arbeitsspeicher der geöffneten Seite und beginnt nach einem Neuladen erneut leer.

Zeilen und Spalten

Zeilen und Spalten können eingefügt und gelöscht werden.

Tables führt dabei betroffene Formelbezüge automatisch nach – sowohl:

- innerhalb desselben Arbeitsblatts,
- als auch in anderen Arbeitsblättern.

Auch:

- Zellbereiche,
- Diagrammbereiche,
- Regeln der bedingten Formatierung,

- verbundene Zellen

werden entsprechend angepasst.

Bezüge auf tatsächlich gelöschte Zellen werden zu:

#BEZUG!

Sortieren, Suchen und Ersetzen

Markierte Tabellenbereiche können auf- oder absteigend sortiert werden.

Dabei werden die vollständigen Zeilen einschließlich ihrer Formatierungen gemeinsam umsortiert.

Die integrierte Suche arbeitet innerhalb des aktiven Arbeitsblatts.

Gesucht werden Zellinhalte beziehungsweise bei Formeln der Formeltext.

Treffer können einzeln oder vollständig ersetzt werden.

Suchen und Ersetzen kann optional zwischen Groß- und Kleinschreibung unterscheiden.

Zellformatierung

Tables unterstützt unter anderem:

- Fett,
- Kursiv,
- Unterstrichen,
- Textfarbe,
- Hintergrundfarbe,
- horizontale Ausrichtung,
- vertikale Ausrichtung,
- Rahmen,
- verbundene Zellen.

Als Zahlenformate stehen zur Verfügung:

- Standard,
- Ganzzahl,
- Dezimal mit zwei Nachkommastellen,
- Währung €,
- Prozent,
- Datum.

Die Formatierung verändert nur die Darstellung. Der eigentliche Zellwert bleibt für Berechnungen unverändert.

Bedingte Formatierung

Zellen können abhängig von ihrem Inhalt automatisch mit Text- und Hintergrundfarben versehen werden.

Unterstützte Bedingungen sind unter anderem:

- größer als,
- kleiner als,
- größer oder gleich,
- kleiner oder gleich,
- gleich,
- ungleich,
- zwischen,
- enthält Text,
- ist Fehler.

Treffen mehrere Regeln auf dieselbe Zelle zu, gilt die zuletzt angelegte Regel.

Die bedingte Formatierung wird im Druckbericht dargestellt, jedoch nicht in eine XLSX-Datei exportiert.

Fixieren von Zeilen und Spalten

Zeilen oberhalb und Spalten links einer ausgewählten Zelle können fixiert werden.

Dadurch bleiben beispielsweise:

- Tabellenüberschriften,
- Beschriftungsspalten

beim Scrollen sichtbar.

Die Fixierung wird für jedes Arbeitsblatt separat gespeichert.

Sie ist Bestandteil des nativen `.jbtab`-Formats, wird jedoch nicht nach XLSX exportiert und im Druckbericht nicht berücksichtigt.

Diagramme

Tables unterstützt drei Diagrammtypen:

- Balkendiagramm,
- Liniendiagramm,

- Kreisdiagramm.

Diagramme werden aus einem markierten Datenbereich erzeugt.

Sie bleiben mit ihrem Datenbereich verbunden und werden bei Änderungen der zugrunde liegenden Werte entsprechend aktualisiert.

Diagramme können:

- verschoben,
- in der Größe verändert,
- zwischen den unterstützten Diagrammtypen umgeschaltet,
- gelöscht

werden.

Beim Einfügen oder Löschen von Zeilen und Spalten werden die zugrunde liegenden Diagrammbereiche mitgeführt.

Diagramme werden im nativen `.jbtab`-Format und im Druckbericht gespeichert beziehungsweise ausgegeben.

Beim XLSX-Export werden die Diagramme selbst nicht übertragen; die zugrunde liegenden Werte bleiben erhalten.

Daten und Datenaustausch

Natives `.jbtab`-Format

Das native und verlustfreie Speicherformat von Tables ist:

.jbtab

Der Dateiname folgt dem Schema:

JJJJMMTT_HHMMSS_table.jbtab

Die Datei ist ein ZIP-Container mit einer internen:

workbook.json

Gespeichert werden die vollständige Arbeitsmappe und unter anderem:

- alle Arbeitsblätter,
- Werte,
- Formeln,
- Formatierungen,
- Diagramme,
- Regeln der bedingten Formatierung,

- Fixierungen.

Zusätzlich enthält die Datei Angaben zu:

- Werkzeug,
- Version,
- Exportzeitpunkt,
- SHA-256-Prüfsumme.

Ein erfolgreicher `.jbtab`-Import ersetzt die aktuell geöffnete Arbeitsmappe vollständig.

Der Import kann jedoch über die Rückgängig-Funktion zurückgenommen werden, solange die Seite geöffnet bleibt.

XLSX

Tables kann XLSX-Dateien exportieren.

Übertragen werden:

- alle Arbeitsblätter,
- Zellwerte,
- Formeln,
- Spaltenbreiten,
- verbundene Zellen,
- Zellformatierungen einschließlich Datumsformat.

Die Formeln werden beim Export in eine für Excel erwartete Schreibweise übersetzt.

Dadurch können beispielsweise Microsoft Excel, LibreOffice Calc und vergleichbare Programme die Formeln selbst neu berechnen.

Beim XLSX-Import werden die enthaltenen Arbeitsblätter an die vorhandene Arbeitsmappe **angehängt**.

Unterstützt werden:

- `.xlsx`,
- `.xls`,
- `.xlsm`.

Existiert ein Blattname bereits, erhält das importierte Arbeitsblatt einen Zusatz wie:

„(2)“

Interne Bezüge der importierten Datei werden entsprechend auf die umbenannten Blätter umgeleitet.

Nicht über XLSX übertragen werden:

- Diagramme,
- bedingte Formatierung,
- Fixierungen.

CSV

CSV-Import und CSV-Export betreffen ausschließlich das aktive Arbeitsblatt.

CSV überträgt nur Werte und keine:

- Formeln,
- Formatierungen,
- Diagramme.

Der Export verwendet das Semikolon als Trennzeichen.

Beim Import erkennt Tables Semikolon oder Komma automatisch und erzeugt aus den Daten ein neues Arbeitsblatt.

CSV ist damit ein einfaches Austauschformat, aber **kein vollständiges Sicherungsformat für eine Tables-Arbeitsmappe**.

Datenintegrität

Beim Export einer `.jbtab`-Datei berechnet Tables einen **SHA-256-Hash über Programmversion und Blattdaten der gesamten Arbeitsmappe**.

Da auch die Programmversion in die Berechnung eingeht, erzeugt derselbe Tabelleninhalt unter einer anderen Programmversion einen anderen Hash.

Beim Import wird die Prüfsumme erneut berechnet.

Tables unterscheidet unter anderem:

„✓ Hash gültig“
Die Datei ist seit dem Export unverändert.

„– Kein Hash vorhanden“
Die Datei enthält keinen entsprechenden Prüfwert.

„⚠ Hash-Abweichung“
Inhalt oder Version stimmen nicht mehr mit dem beim Export berechneten Zustand überein.

Eine Datei mit Hash-Abweichung wird trotzdem geladen. Zusätzlich erscheint eine Warnung.

Nach einer Änderung der Arbeitsmappe ist ein zuvor geprüfter Hash nicht mehr für den aktuellen Arbeitsstand gültig.

Tables kennzeichnet dies entsprechend.

Der SHA-256 dient bei Tables dem Nachweis der Unveränderheit der Arbeitsmappe. Ein Datenaustausch mit der `project.dbm` findet dadurch nicht statt.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Tables läuft vollständig lokal im Browser als einzelne HTML-Datei.

Es benötigt:

- keine Cloud,
- keinen Server,
- kein Benutzerkonto.

Änderungen werden ungefähr eine halbe Sekunde nach der letzten Eingabe automatisch im lokalen Browserspeicher gesichert.

Auch beim Schließen oder Neuladen wird der aktuelle Stand noch gespeichert.

Beim nächsten Öffnen im selben Browser wird dieser Arbeitsstand automatisch wiederhergestellt.

Damit steht innerhalb von JUMBO BLOCK Rescue eine allgemeine Tabellenkalkulation zur Verfügung, ohne dass eine externe Office-Suite oder ein Cloud-Dienst für grundlegende Tabellen-, Rechen- und Auswertungsaufgaben erforderlich ist.

Datensicherung

Der automatische Browserspeicher ist an:

- das verwendete Gerät,
- den verwendeten Browser

gebunden.

Er ist deshalb nicht als alleinige dauerhafte Datensicherung vorgesehen.

Für eine vollständige und weitergebbare Sicherung sollte die Arbeitsmappe regelmäßig als:

.jbtab

exportiert werden.

Dieses Format ist das native verlustfreie Sicherungsformat von Tables.

XLSX eignet sich für den Austausch mit anderen Tabellenkalkulationen, überträgt aber nicht alle Tables-Funktionen.

CSV enthält ausschließlich Werte des aktiven Arbeitsblatts und ist deshalb nicht als vollständiges Sicherungsformat geeignet.

Drucken und PDF

Über **„Druck / PDF“** erzeugt Tables einen formatierten Bericht.

Der Bericht enthält:

- den gewählten Druckbereich,

- die Tabelle mit tatsächlichen Spaltenbreiten,
- bedingte Formatierungen,
- Diagramme des Arbeitsblatts,
- Werkzeugversion,
- Erstellungszeitpunkt,
- SHA-256-Prüfsumme der Arbeitsmappe.

Vor dem Erzeugen des Berichts wird die Arbeitsmappe automatisch neu berechnet.

Als Papierformate stehen zur Verfügung:

- A4,
- A3,
- A2.

Die Ausgabe kann im:

- Querformat,
- Hochformat

erfolgen.

Querformat ist die Voreinstellung.

Zusätzlich kann festgelegt werden, ob Hilfslinien der Tabelle im Bericht dargestellt werden.

Ist die Arbeitsmappe seit dem letzten `.jbtab`-Export unverändert, entspricht die Prüfsumme des Berichts dem Hash der exportierten Datei. Datei und Ausdruck können dadurch demselben Datenstand zugeordnet werden.

Sprache

Tables unterstützt Deutsch und Englisch über das globale Menü der Engineering Suite.

Dabei werden unter anderem umgeschaltet:

- Beschriftungen,
- Hinweise,
- Meldungen,
- Werkzeugleisten,
- Diagrammbeschriftungen,
- Funktionsnamen im Funktionsmenü,
- Druckbericht.

Der eigentliche Inhalt der Arbeitsmappe wird durch den Sprachwechsel nicht verändert.

Deutsche und englische Funktionsnamen bleiben unabhängig von der Oberflächensprache gültig.

Bei Dezimalzahlen innerhalb von Formeln ist beim Wechsel zwischen den Sprachen jedoch zu beachten, dass die Schreibweise mit Dezimalpunkt in beiden Sprachfassungen funktioniert.

Grenzen und Hinweise

Tables ist eine eigenständige lokale Tabellenkalkulation und tauscht keine Daten mit der gemeinsamen `project.dbm` aus.

Für die Datensicherung gelten unterschiedliche Funktionsumfänge:

`.jbtab` → vollständiges natives Speicherformat

XLSX → Austauschformat mit Formeln und Zellformatierung, jedoch ohne Diagramme, bedingte Formatierung und Fixierungen

CSV → ausschließlich Werte des aktiven Arbeitsblatts

Der Rückgängig-/Wiederholen-Verlauf bleibt nur erhalten, solange die Seite geöffnet ist.

Beim Ausschneiden und anschließenden Einfügen werden Formeln an anderen Stellen, die auf die ausgeschnittenen Zellen verweisen, nicht angepasst.

Die Ausfüllfunktion erzeugt keine automatischen Zahlenreihen, sondern überträgt Inhalt beziehungsweise Formeln der ersten Zeile oder Spalte.

Suchen und Ersetzen arbeitet nur im aktiven Arbeitsblatt.

Das Feld **„Dein Name (für Protokoll)“** ist nach dem Handbuch derzeit ausschließlich ein Notizfeld. Es wird weder in den Export noch in den Druckbericht oder die SHA-256-Prüfsumme aufgenommen.

Technische Einordnung in JUMBO BLOCK Rescue

Tables erweitert JUMBO BLOCK Rescue um eine **vollständig lokale Tabellenkalkulation mit eigenem Formelmotor und nativer Dateisicherung**.

Die funktionale Zuordnung lautet:

Calc → schnelle Einzel- und Hilfsberechnungen

Units → Einheitenumrechnung

Tables → strukturierte Tabellen, Formeln, Listen, Auswertungen und Diagramme

Durch XLSX- und CSV-Unterstützung kann Tables zusätzlich als lokales Austauschwerkzeug für vorhandene Tabellendaten dienen.

Das native `.jbtab`-Format erhält dagegen den vollständigen Tables-Arbeitsstand einschließlich der programmspezifischen Funktionen und SHA-256-Integritätsinformationen.

Damit können während eines IT-Ausfalls Tabellen, technische Berechnungen, Listen und Auswertungen vollständig lokal erstellt, fortgeführt, gesichert und als PDF dokumentiert werden.

Weiterführende Dokumentation

JUMBO BLOCK Tables – Benutzerhandbuch · Version 2.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Tables“ auf **https://jumboblock.org/**.

D.13 – Writer

Bereich: Dokumente
Programm: JUMBO BLOCK Writer
Baseline 20261001: 10.1.0 ff.

Zweck

JUMBO BLOCK Writer ist die vollständig lokal und offline arbeitende Textverarbeitung der JUMBO BLOCK Engineering Suite.

Das Programm läuft als einzelne HTML-Datei im Browser und benötigt:

- keinen Server,
- keine Cloud,
- kein Benutzerkonto,
- keinen Netzwerkzugriff.

Writer ermöglicht die Erstellung und Bearbeitung formatierter Dokumente einschließlich Tabellen, Bildern, Überschriften, Inhaltsverzeichnis, automatischen Feldern und Seitenlayout.

Dokumente können unter anderem als:

- Writer-`.dbm`,
- Word-Datei `.docx`,
- PDF

ausgegeben werden.

Word-Dateien können wieder importiert werden. Zusätzlich steht ein eingeschränkter ODT-Import zur Verfügung.

Damit stellt Writer innerhalb von JUMBO BLOCK Rescue eine eigenständige lokale Textverarbeitung bereit.

Rolle innerhalb der Engineering Suite

Writer übernimmt die **allgemeine Dokumenterstellung und Textverarbeitung** innerhalb der Engineering Suite.

Es ist nicht an das gemeinsame `project.dbm`-Datenmodell der projektspezifischen Engineering-Anwendungen gebunden.

Writer eignet sich beispielsweise für:

- technische Dokumentationen,
- Berichte,

- Geschäftsbriefe,
- Aktenvermerke,
- Protokolle,
- freie Projektdokumente.

Die funktionale Trennung lautet:

Project → strukturiertes Projektmanagement und Projektdokumentation

Note → freie Notizen und ergänzende Informationen

Tables → Tabellen und Auswertungen

Writer → formatierte Dokumente und Textverarbeitung

Dokumentaufbau

Ein Writer-Dokument besteht aus vier unabhängig editierbaren Bereichen:

Briefkopf

Der Briefkopf erscheint ausschließlich auf der ersten Seite.

Er kann beispielsweise für:

- Absenderangaben,
- Logo,
- Anschrift

verwendet werden.

Haupttext

Der Haupttext bildet den fortlaufenden Textkörper des Dokuments und wird automatisch über die benötigten Seiten verteilt.

Fußfeld

Das Fußfeld erscheint ausschließlich auf der letzten Seite.

Es eignet sich beispielsweise für:

- Grußformeln,
- Unterschriftenbereiche.

Seitenfuß

Der Seitenfuß wird auf jeder Seite wiederholt und kann beispielsweise Firmenangaben oder Datumsinformationen enthalten.

Unterhalb dieses Bereichs erzeugt Writer im PDF zusätzlich eine eigene Prüfzeile mit SHA-256-Prüfwert und Seitenzahl.

Briefkopf, Fußfeld und Seitenfuß können ein- und ausgeblendet werden. Der Haupttext bleibt immer aktiv.

Textformatierung

Writer stellt eine umfangreiche Formatierungsleiste bereit.

Unterstützt werden unter anderem:

- Fett,
- Kursiv,
- Unterstrichen,
- Durchgestrichen,
- Hochgestellt,
- Tiefgestellt,
- Textfarbe,
- Hintergrundfarbe,
- Entfernen von Formatierungen,
- linksbündig,
- zentriert,
- rechtsbündig,
- Blocksatz,
- Aufzählungen,
- Nummerierungen,
- Links,
- horizontale Trennlinien,
- Seitenumbrüche.

Als Absatzformate stehen zur Verfügung:

- Standard,
- Überschrift 1,
- Überschrift 2,
- Überschrift 3.

Die Schriftgröße kann von **6 bis 68 pt** eingestellt werden.

Für die Zeilenhöhe stehen Werte von **0,8 bis 3,0** sowie eine Standardeinstellung zur Verfügung.

Schriftarten

Writer enthält zwölf lokal mitgelieferte Schriftarten.

Serifenlos

- Liberation Sans,
- DejaVu Sans,
- Noto Sans,
- Ubuntu,
- Cantarell.

Serifen

- Liberation Serif,
- DejaVu Serif,
- Noto Serif,
- Nimbus Roman.

Monospace

- Liberation Mono,
- DejaVu Sans Mono,
- Noto Sans Mono,
- Ubuntu Mono.

Da die Schriften lokal bereitgestellt werden, ist für ihre Verwendung keine Online-Verbindung erforderlich.

Überschriften und Inhaltsverzeichnis

Überschriften werden über die Absatzformate:

- Überschrift 1,
- Überschrift 2,
- Überschrift 3

definiert.

Writer kann daraus im Haupttext automatisch ein Inhaltsverzeichnis erzeugen.

Das Inhaltsverzeichnis berücksichtigt die Überschriften entsprechend ihrer Gliederungsebene und enthält:

- Überschriftentext,

- Einrückung nach Ebene,
- Punktlinie,
- Seitenzahl.

Neue, geänderte oder entfernte Überschriften werden automatisch berücksichtigt.

Beim PDF-Export werden die Seitenzahlen anhand der tatsächlichen PDF-Seitenaufteilung eingesetzt.

Überschriften aus Briefkopf, Fußfeld und Seitenfuß werden nicht in das Inhaltsverzeichnis aufgenommen.

Automatische Felder

Writer unterstützt automatische Felder für:

- Datum,
- Uhrzeit,
- Seitenzahl,
- Gesamtseitenzahl.

Datum und Uhrzeit werden beim Laden des Dokuments aktualisiert.

Seitenzahl und Gesamtseitenzahl erhalten ihre endgültigen Werte bei der Ausgabe.

Im Word-Export werden Seitenzahlen als Word-Felder geschrieben und anschließend von Word selbst berechnet.

Tabellen

Writer kann Tabellen direkt in das Dokument einfügen.

Beim Erstellen werden Zeilen- und Spaltenzahl festgelegt.

Anschließend können unter anderem:

- Zeilen oberhalb oder unterhalb eingefügt,
- Zeilen gelöscht,
- Spalten links oder rechts eingefügt,
- Spalten gelöscht,
- Zellinhalte vertikal ausgerichtet,
- Zellrahmen ein- oder ausgeblendet,
- vollständige Tabellen gelöscht

werden.

Spaltenbreiten können direkt durch Ziehen verändert werden.

Rahmenlose Tabellen können beispielsweise für die Positionierung von Inhalten im Briefkopf verwendet werden.

Bilder

Bilddateien können direkt in ein Writer-Dokument eingebettet werden.

Die Bilddaten werden dabei als Base64-Daten Bestandteil des Dokuments. Es entstehen keine externen Bildverweise.

Bilder können:

- proportional skaliert,
- verschoben,
- zentriert,
- ohne Textumfluss,
- links umflossen,
- rechts umflossen

dargestellt werden.

Zusätzlich kann die vertikale Ausrichtung zur Textzeile eingestellt werden.

Links und Dateien

Unterhalb des eigentlichen Dokuments besitzt Writer einen separaten Bereich:

„Links und Dateien“

Ein Eintrag besteht aus:

- Bezeichnung,
- Ziel.

Als Ziel kann beispielsweise eine Webadresse oder ein relativer Dateipfad verwendet werden.

Die Funktion **„Datei verknüpfen“** übernimmt bei einer lokalen Datei deren Namen als Ausgangspunkt.

Im Unterschied zu Bildern wird eine auf diese Weise verknüpfte Datei **nicht in das Writer-Dokument eingebettet**. Gespeichert wird lediglich der Verweis.

Relative Dateipfade können auch bei der Nutzung von einem USB-Stick verwendet werden.

Seiteneinstellungen

Writer unterstützt:

Papierformate

- A4,
- Letter.

Ausrichtung

- Hochformat,
- Querformat.

Zusätzlich können die vier Seitenränder individuell in Millimetern eingestellt werden.

Die Standardeinstellung lautet:

- oben 10 mm,
- links 25 mm,
- rechts 10 mm,
- unten 10 mm.

Die Seitenaufteilung wird anhand dieser Einstellungen berechnet.

Vorlagen

Writer enthält vorbereitete Dokumentvorlagen.

Geschäftsbrief

Die Geschäftsbriefvorlage ist an DIN 5008 angelehnt und enthält unter anderem:

- Firmenname,
- Rücksendeangabe,
- Anschriftfeld,
- Informationsblock,
- Zeichen,
- Ansprechpartner,
- Datumsfeld,
- Betreff,
- Anrede,
- Grußformel,
- Unterschrift,
- Firmenangaben im Seitenfuß.

Die exakten Maße der DIN 5008 – beispielsweise für Fensterumschläge – müssen vor einer entsprechenden Verwendung gesondert geprüft werden.

Aktenvermerk

Die Vorlage enthält unter anderem eine Kopftabelle für:

- Datum,
- Verfasser,
- Aktenzeichen,
- Betreff,
- Verteiler

sowie eine vorbereitete Gliederung.

Protokoll

Die Protokollvorlage enthält unter anderem:

- Anlass,
- Datum und Uhrzeit,
- Ort,
- Teilnehmende,
- Protokollführung,
- Tagesordnung,
- Tagesordnungspunkte,
- Maßnahmentabelle mit Verantwortlichkeit und Termin.

Die Vorlagen werden in der aktuell eingestellten Oberflächensprache erzeugt.

Suchen und Ersetzen

Writer besitzt eine dokumentweite Such- und Ersetzen-Funktion.

Durchsucht werden die eingeblendeten Bereiche:

- Briefkopf,
- Haupttext,
- Fußfeld,
- Seitenfuß.

Automatische Felder und das Inhaltsverzeichnis werden übersprungen.

Die Suche kann optional zwischen Groß- und Kleinschreibung unterscheiden.

Treffer können einzeln oder vollständig ersetzt werden.

Wörter- und Zeichenzählung

Writer zeigt fortlaufend:

- Wörter,
- Zeichen mit Leerzeichen,
- Zeichen ohne Leerzeichen

des Haupttexts an.

Ist Text markiert, wird zusätzlich die Wortzahl der Auswahl angezeigt.

Das Inhaltsverzeichnis wird bei der Zählung nicht berücksichtigt.

Daten und Datenaustausch

Writer verwendet für die vollständige Sicherung des eigenen Dokumentzustands ein `.dbm`-Format.

Über **„Dokument exportieren“** werden unter anderem gespeichert:

- Briefkopf,
- Haupttext,
- Fußfeld,
- Seitenfuß,
- Formatierungen,
- eingebettete Bilder,
- Inhaltsverzeichnis,
- Seiteneinstellungen,
- Aktivierungszustand der Dokumentbereiche,
- Links und Dateien,
- Programmversion,
- SHA-256-Prüfwert.

Die Writer-`.dbm` ist ein lesbares JSON-Format.

Writer verwendet damit ein eigenes Dokumentformat und nicht die gemeinsame `project.dbm` der projektspezifischen Engineering-Anwendungen.

Import und Sicherheitsbereinigung

Writer kann eigene `.dbm`-Dokumente wieder vollständig importieren.

Beim Import werden die Inhalte zusätzlich bereinigt.

Übernommen werden nur:

- Elemente,
- Attribute,
- Formatierungen,

die Writer selbst unterstützt beziehungsweise erzeugt.

Entfernt werden unter anderem:

- Skripte,
- Ereignisattribute,
- fremde eingebettete Inhalte,
- externe Bilder,
- unsichere Links.

Wurden Inhalte bei der Bereinigung entfernt, zeigt Writer dies in der Statuszeile an.

Damit wird verhindert, dass ein importiertes Dokument ungeprüft aktive oder nicht unterstützte Inhalte in den Writer übernimmt.

Word-Export

Writer kann Dokumente als:

.docx

exportieren.

Übertragen werden unter anderem:

- Textformatierungen,
- Schriftarten und Schriftgrößen,
- Absatzformatierungen,
- Überschriften als Word-Formatvorlagen,
- Aufzählungen,
- Nummerierungen,
- Tabellen,
- Bilder,
- Links,
- Trennlinien,
- Seitenumbrüche,
- Briefkopf,

- Seitenfuß,
- Fußfeld,
- automatische Felder,
- Inhaltsverzeichnis,
- Papierformat,
- Ausrichtung,
- Seitenränder.

Der SHA-256-Prüfwert wird nicht in die Word-Datei übernommen, da ein anschließend in Word bearbeitetes Dokument nicht mehr dem geprüften Writer-Datenstand entspricht.

Word-Import

Writer kann `.docx`-Dateien beispielsweise aus Microsoft Word oder LibreOffice Writer importieren.

Übernommen werden unter anderem:

- wesentliche Textformatierungen,
- Überschriften,
- Aufzählungen und Nummerierungen,
- Tabellen,
- eingebettete PNG-, JPEG- und GIF-Bilder,
- Weblinks,
- Seitenumbrüche,
- Trennlinien,
- Kopfzeile,
- Fußzeile,
- automatische Felder,
- Dokumenttitel,
- Papierformat,
- Ausrichtung,
- Seitenränder.

Bei Dokumenten, die ursprünglich aus Writer exportiert wurden, kann Writer das am Ende des Haupttexts platzierte Fußfeld wieder als eigenen Dokumentbereich herstellen.

Nicht übernommen werden unter anderem:

- schwebende Textfelder und Formen,
- Fußnoten,
- Endnoten,
- Kommentare,
- Formeln,
- eingebettete Objekte,
- verknüpfte, nicht eingebettete Bilder,
- mehrspaltiger Satz,
- zusätzliche Kopf- und Fußzeilenvarianten.

Nicht übernommene Inhalte werden nach dem Import in der Statuszeile gemeldet.

Aktive oder unsichere Inhalte werden nicht übernommen.

ODT-Import

Writer kann außerdem `.odt`-Dateien importieren.

Dabei wird jedoch ausschließlich der **reine Absatztext** in den Haupttext übernommen.

Nicht übernommen werden:

- Formatierungen,
- Bilder,
- zusätzliche Writer-Dokumentbereiche,
- Prüfwerte.

Der ODT-Import ist damit ausdrücklich eingeschränkter als der DOCX-Import.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Writer läuft vollständig lokal als einzelne HTML-Datei.

Alle benötigten Bibliotheken sind in die Anwendung eingebettet.

Dadurch funktioniert Writer auch:

- ohne Netzwerkzugriff,
- direkt von einem USB-Stick,
- während eines Ausfalls zentraler IT-Infrastruktur.

Änderungen werden ungefähr eine halbe Sekunde nach der letzten Eingabe automatisch im lokalen Browserspeicher gesichert.

Beim erneuten Öffnen im selben Browser wird der zuletzt gespeicherte Stand wieder geladen.

Damit kann JUMBO BLOCK Rescue neben Diagnose-, Rettungs- und Engineering-Aufgaben auch als lokaler Arbeitsplatz für die Erstellung und Fortführung umfangreicher Dokumente eingesetzt werden.

Datensicherung

Der automatische Browser-Speicher ist an den jeweiligen Browser und das verwendete Gerät gebunden.

Für eine dauerhafte und übertragbare Sicherung sollte das Dokument deshalb als Writer-`.dbm` exportiert werden.

„Bildschirm löschen“ entfernt den im Browser gespeicherten Dokumentstand nach einer Sicherheitsabfrage endgültig.

Bereits exportierte:

- `.dbm`,
- `.docx`,
- PDF-Dateien

werden dadurch nicht verändert.

PDF-Export

Writer erzeugt die PDF-Datei vollständig lokal im Browser.

Ein separater Druckdialog ist dafür nicht erforderlich.

Beim Export verteilt Writer den Haupttext entsprechend der eingestellten Seitengröße und Ränder auf die erforderlichen Seiten.

Dabei gelten:

- Briefkopf nur auf der ersten Seite,
- Fußfeld nur auf der letzten Seite,
- Seitenfuß auf jeder Seite,
- manuelle Seitenumbrüche werden berücksichtigt,
- lange Listen und Tabellen werden über mehrere Seiten verteilt.

Seitenzahlfelder und Seitenzahlen im Inhaltsverzeichnis erhalten bei der PDF-Erzeugung ihre tatsächlichen Werte.

Die Seiten werden beim PDF-Export als Bild in das PDF übernommen.

Der Text ist deshalb im erzeugten PDF **nicht durchsuchbar und nicht kopierbar**.

Datenintegrität

Writer verwendet **SHA-256** zur Integritätsprüfung seiner eigenen Dokumente.

Beim Export einer `.dbm` wird ein SHA-256-Prüfwert über den vollständigen Dokumentinhalt einschließlich der Programmversion gebildet.

Beim PDF-Export wird derselbe Fingerabdruck erzeugt.

Am unteren Rand jeder PDF-Seite erscheint eine Prüfzeile:

links → vollständiger SHA-256-Wert

rechts → Seite x / y

Damit kann ein PDF-Ausdruck eindeutig einem unveränderten Writer-`.dbm`-Datenstand zugeordnet werden.

Beim Import einer Writer-`.dbm` unterscheidet das Programm:

- **Hash gültig**
- **Kein Hash vorhanden**
- **Hash-Abweichung**

Die SHA-256-Prüfung dient zur Erkennbarkeit nachträglicher Veränderungen.

Sie ersetzt ausdrücklich **keine qualifizierte elektronische Signatur (QES) nach eIDAS**.

Ansicht und Bedienung

Writer kann die Dokumentansicht an die verfügbare Fensterbreite anpassen.

Der Zoom kann in 10-Prozent-Schritten verändert werden.

100 % stellt die reale Papiergröße dar.

Zusätzlich steht ein Vollbildmodus zur Verfügung.

Für wichtige Bearbeitungsfunktionen existieren Tastenkürzel, unter anderem:

- `Strg + Z` → Rückgängig,
- `Strg + Y` → Wiederholen,
- `Strg + F` → Suchen.

Sprache

Writer unterstützt Deutsch und Englisch.

Die Umschaltung erfolgt über das globale Menü der Engineering Suite.

Umgeschaltet werden unter anderem:

- Beschriftungen,
- Hinweise,
- Platzhalter,

- Kontextwerkzeuge,
- Statusmeldungen,
- Prüfwertmeldungen,
- Inhaltsverzeichnis,
- Beschriftungen der Seitenumbrüche.

Der eigentliche Dokumentinhalt wird nicht automatisch übersetzt.

Vorlagen werden in der Sprache erzeugt, die beim Laden der jeweiligen Vorlage eingestellt ist.

Grenzen und Hinweise

Writer ist eine lokale Textverarbeitung mit eigenem Dokumentmodell.

Die Bildschirmdarstellung zeigt Briefkopf, Fußfeld und Seitenfuß jeweils nur einmal. Die tatsächliche Wiederholung beziehungsweise Seitenzuordnung wird erst in der exportierten Datei sichtbar.

Seitenzahlfelder erhalten ihren endgültigen Wert erst beim PDF- beziehungsweise Word-Export.

Die SHA-256-Prüfzeile erscheint auf jeder von Writer erzeugten PDF-Seite und kann nicht abgeschaltet werden.

Importierte Dateien werden bewusst auf die von Writer unterstützten Inhalte reduziert. Nicht unterstützte oder potenziell unsichere Bestandteile gehen dabei verloren.

Beim Word-Export werden Bilder ohne Textumfluss platziert.

Fußnoten werden derzeit nicht unterstützt.

Der Word-Import übernimmt unter anderem keine Textfelder, Fußnoten, Kommentare, Formeln oder eingebetteten Objekte.

Der ODT-Import beschränkt sich auf reinen Absatztext.

Die SHA-256-Integritätsprüfung weist die Unveränderheit eines Writer-Datenstands nach, stellt aber keine elektronische Signatur dar.

Technische Einordnung in JUMBO BLOCK Rescue

Writer erweitert JUMBO BLOCK Rescue um eine **vollständig lokale, formatierte Textverarbeitung und Dokumentenerstellung**.

Die funktionale Zuordnung lautet:

Note → schnelle freie Notizen

Tables → Tabellen, Formeln und Auswertungen

Writer → umfangreiche formatierte Dokumente, Berichte, Briefe und Protokolle

Durch die lokale Browser-Speicherung und das eigene `.dbm`-Format können Dokumente unabhängig von einer zentralen Office-Infrastruktur erstellt und gesichert werden.

Der DOCX-Import und -Export ermöglicht zusätzlich den Austausch mit verbreiteten Textverarbeitungen.

Die SHA-256-Verknüpfung zwischen Writer-`.dbm` und PDF-Ausgabe schafft darüber hinaus eine nachvollziehbare Zuordnung zwischen bearbeitbarem Dokumentstand und ausgegebenem PDF.

Damit kann JUMBO BLOCK Rescue auch bei einem Ausfall der regulären IT-Infrastruktur als eigenständiger lokaler Arbeitsplatz für wesentliche Dokumentations- und Textverarbeitungsaufgaben eingesetzt werden.

Weiterführende Dokumentation

JUMBO BLOCK Writer – Benutzerhandbuch · Version 10.1.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Writer“ auf **https://jumboblock.org/.**

D.14 – Reader

Bereich: Dokumente
Programm: JUMBO BLOCK Reader
Baseline 20261001: 1.0.4 ff.

Zweck

JUMBO BLOCK Reader ist der lokale Dokumenten- und E-Book-Reader der JUMBO BLOCK Engineering Suite.

Er ist sowohl für technische Dokumentationen und Publikationen als auch für allgemeine E-Books ausgelegt.

Unterstützt werden:

- EPUB,
- PDF,
- Markdown,
- TXT,
- HTML/HTM,
- JSON,
- XML.

Reader läuft vollständig lokal im Browser. Dokumente werden nicht zu einem Server hochgeladen, und für die eigentliche Dokumentanzeige ist kein Benutzerkonto erforderlich.

Die Anwendung ist für:

- Desktop,
- Tablet,
- Smartphone

ausgelegt.

Rolle innerhalb der Engineering Suite

Reader übernimmt innerhalb der Engineering Suite die **lokale Dokumenten- und Leseebene**.

Die funktionale Abgrenzung lautet:

Writer → Dokumente erstellen und bearbeiten

Reader → vorhandene Dokumente und E-Books lesen und navigieren

Reader ist damit kein Textverarbeitungsprogramm und verändert die geöffneten Dokumente nicht.

Stattdessen stellt das Programm eine einheitliche lokale Leseoberfläche für unterschiedliche Dokument- und Textformate bereit.

Es eignet sich insbesondere für:

- Handbücher,
- technische Dokumentationen,
- PDF-Unterlagen,
- E-Books,
- Markdown-Dokumente,
- Textdateien,
- lokale HTML-Dokumente,
- strukturierte JSON- und XML-Dateien.

Unterstützte Dateiformate

EPUB

EPUB-Dateien werden als E-Books dargestellt.

Unterstützt werden unter anderem:

- echtes EPUB-Inhaltsverzeichnis,
- Kapitel,
- interne Links,
- Bilder,
- Lesezeichen,
- Textmarkierungen.

PDF

PDF-Dokumente werden seitenweise dargestellt.

Zur Verfügung stehen:

- Seitennavigation,
- Zoom,
- Textsuche.

Markdown

Unterstützt werden:

- `.md`,
- `.markdown`.

Markdown-Dokumente werden in einer formatierten Leseansicht dargestellt, die insbesondere Überschriften, Hervorhebungen und einfache Markdown-Strukturen berücksichtigt.

Text

TXT-Dateien werden als fortlaufender skalierbarer Text dargestellt.

HTML/HTM

Lokale HTML-Dokumente werden in einer **bereinigten Leseansicht** angezeigt.

JSON und XML

JSON- und XML-Dateien werden als strukturierter Text dargestellt.

Diese Funktion ist insbesondere für technische Daten- und Austauschdateien nützlich.

Dokument öffnen

Über:

„ Öffnen“

wird der Dateiauswahldialog des Betriebssystems aufgerufen.

Nach dem Laden zeigt Reader den Dateinamen im Kopfbereich an.

Status und Lesefortschritt werden am unteren Rand dargestellt.

Reader merkt sich lokal:

- die zuletzt geöffnete Datei,
- dokumentbezogene Einstellungen,
- Lesepositionen.

Beim nächsten Start versucht Reader, die zuletzt gelesene Datei erneut zu öffnen und an der gespeicherten Position fortzusetzen.

Ob der Browser die lokale Datei dauerhaft wieder bereitstellen kann, hängt von dessen lokalen Speicher- und Sicherheitsregeln ab.

Dokumentbezogene Einstellungen

Für jedes Dokument werden unter anderem getrennt gespeichert:

- Leseposition,
- Schriftgröße beziehungsweise PDF-Zoom,
- Lesebreite,

- Lesezeichen,
- Markierungen.

Damit können unterschiedliche Dokumente mit jeweils eigenen Leseeinstellungen verwendet werden.

Navigation

Die Schaltflächen:

◀ / ▶

dienen zur Navigation.

Ihre Funktion hängt vom geöffneten Dokumenttyp ab.

EPUB

Wechsel zum vorherigen beziehungsweise nächsten Kapitel.

PDF

Wechsel zur vorherigen beziehungsweise nächsten Seite.

Textformate

Scrollen ungefähr eine Bildschirmhöhe zurück beziehungsweise weiter.

Inhaltsverzeichnis und Seitenleiste

Über:

☰

wird die Seitenleiste geöffnet.

EPUB

Bei EPUB zeigt der Bereich **„Inhalt“** das im Buch hinterlegte Inhaltsverzeichnis.

Unterstützt werden:

- EPUB-3-Navigation-Dokumente (`nav.xhtml`),
- EPUB-2-NCX-Inhaltsverzeichnisse.

Verschachtelte Kapitel und Unterkapitel bleiben erhalten.

Fehlt ein echtes Inhaltsverzeichnis, versucht Reader zunächst, Überschriften aus den Kapiteln zu verwenden.

Erst wenn auch dies nicht möglich ist, wird auf eine einfache Kapitelliste zurückgegriffen.

PDF

Bei PDF zeigt die Seitenleiste eine Seitenliste.

Durch Auswahl einer Seite kann unmittelbar zu dieser Seite gesprungen werden.

Interne und externe Links

Interne Links innerhalb eines EPUB werden innerhalb des Readers aufgelöst.

Dies betrifft beispielsweise:

- Inhaltsverzeichnisse innerhalb einer Buchseite,
- Fußnoten,
- Querverweise.

Externe Weblinks werden getrennt behandelt.

Für deren Aufruf kann eine Internetverbindung erforderlich sein.

Darstellung

Bei EPUB und Textformaten verändern:

A− / A+

die Schriftgröße.

Der einstellbare Bereich reicht von:

8 px bis 72 px

Bei PDF verändern dieselben Schaltflächen stattdessen den Zoom.

Lesebreite

Für Textdokumente kann die maximale Lesebreite gewählt werden.

Zur Verfügung stehen:

- 720,
- 900,
- 1100,
- 1400.

Damit kann die Zeilenlänge an unterschiedliche Bildschirmgrößen angepasst werden.

Helle und dunkle Darstellung

Die Schaltfläche:

‹

wechselt zwischen heller und dunkler Darstellung.

Damit kann die Leseansicht an Umgebungslicht und persönliche Präferenzen angepasst werden.

Fokusmodus

Über:

wird der Fokusmodus aktiviert.

Dabei werden:

- Kopfbereich,
- Suite-Menü,
- Statusleiste

ausgeblendet.

Reader nutzt dadurch einen größeren Teil der verfügbaren Browserfläche für das eigentliche Dokument.

Der Fokusmodus arbeitet unabhängig von der Fullscreen-Unterstützung des Browsers.

Vollbildmodus

Über:

fordert Reader den echten Browser-Vollbildmodus an.

Soweit vom Browser unterstützt, können dadurch zusätzlich:

- Browser-Tabs,
- Adressleiste

ausgeblendet werden.

Die tatsächliche Verfügbarkeit hängt vom verwendeten Browser und Gerät ab.

Suche

Das Suchfeld durchsucht das aktuell geöffnete Dokument.

Die Suche wird über:

oder die Eingabetaste gestartet.

PDF

Reader durchsucht den extrahierbaren Text seitenweise und springt zur ersten gefundenen Seite.

EPUB und Textformate

Die Suche arbeitet im aktuell angezeigten Kapitel beziehungsweise Dokumentinhalt.

Ein gefundener Treffer wird über die Suchfunktion des Browsers hervorgehoben.

Grenze der PDF-Suche

Bildbasierte oder eingescannte PDF-Dateien ohne eingebettete Textebene können nicht wie normale Text-PDFs durchsucht werden.

Reader führt keine OCR-Texterkennung durch.

Lesezeichen

Über:

wird an der aktuellen Position ein Lesezeichen gesetzt.

Die gespeicherte Position hängt vom Dokumenttyp ab.

PDF

Gespeichert wird die aktuelle Seite.

EPUB

Gespeichert werden:

- Kapitel,
- relative Leseposition.

Textformate

Gespeichert wird die relative Scrollposition.

Vorhandene Lesezeichen werden im Reiter **„Lesezeichen“** angezeigt und können durch Anklicken erneut aufgerufen werden.

Einzelne Lesezeichen können wieder entfernt werden.

Markierungen

Über:

kann zuvor ausgewählter Text in EPUB- und Textansichten markiert werden.

Die Markierungen werden dokumentbezogen lokal gespeichert.

Beim erneuten Anzeigen des betreffenden Kapitels versucht Reader, die Markierungen wiederherzustellen.

In der Seitenleiste werden vorhandene Markierungen mit einem Textauszug aufgeführt.

Bei mehrfach identischen Textstellen kann die Wiederherstellung eine andere beziehungsweise die erste passende Textstelle treffen.

Markierungen sind deshalb als **persönliche Lesehilfe und nicht als revisionssichere Annotation** einzuordnen.

Tastaturbedienung

Reader kann weitgehend ohne Maus bedient werden.

Solange sich der Eingabefokus nicht in einem Eingabefeld befindet, stehen unter anderem folgende Tasten zur Verfügung:

← / Bild↑
zurück

→ / Bild↓ / Leertaste
weiter

Pos1
zum Anfang

Ende
zum Ende des Dokuments beziehungsweise zum letzten EPUB-Kapitel oder zur letzten PDF-Seite

+ / −
Schriftgröße beziehungsweise PDF-Zoom verändern

F2
Datei öffnen

F3
Suche fokussieren

F4
Seitenleiste öffnen beziehungsweise schließen

F9
Lesezeichen setzen

F10
Fokusmodus

F11
Vollbildmodus

ESC
Fokusmodus beziehungsweise mobile Seitenleiste schließen und gegebenenfalls den Browser-Vollbildmodus beenden.

Funktionstasten können abhängig von Browser oder Betriebssystem bereits anderweitig belegt sein.

Touch-Bedienung

Auf Touch-Geräten unterstützt Reader horizontale Wischbewegungen.

Eine deutliche Wischbewegung:

nach links → weiter

nach rechts → zurück

Auf schmalen Displays passt sich die Oberfläche an.

Die Seitenleiste wird dort als einblendbares Panel dargestellt.

Daten und Datenaustausch

Reader besitzt kein eigenes Dokumentformat für die Bearbeitung der geöffneten Dateien.

Die Originaldokumente bleiben die eigentlichen Datenquellen.

Reader speichert lediglich dokumentbezogene Lesedaten wie:

- Position,
- Anzeigeeinstellungen,
- Lesezeichen,
- Markierungen

lokal im Browser.

Damit unterscheidet sich Reader grundsätzlich von Writer, Tables, Note oder Project, die eigene Arbeits- beziehungsweise Austauschdateien erzeugen.

Lokale Speicherung

Reader arbeitet nach einem **Offline-first-Prinzip**.

Die Dokumentverarbeitung erfolgt lokal auf dem verwendeten Gerät.

Im lokalen Browserspeicher werden unter anderem gespeichert:

- Lesepositionen,
- Anzeigeeinstellungen,
- Lesezeichen,
- Markierungen.

Die zuletzt geöffnete Datei wird – soweit dies vom Browser erlaubt wird – zusätzlich lokal über:

IndexedDB

für ein späteres Wiederöffnen vorgehalten.

Das Löschen von:

- Browserdaten,
- Website-Daten,
- lokalem Speicher

kann diese Informationen und gegebenenfalls auch die vorgehaltene zuletzt gelesene Datei entfernen.

Datenschutz

Reader selbst lädt Dokumentinhalte nicht auf einen Server hoch.

Die eigentliche Dokumentverarbeitung erfolgt lokal.

Eine Netzwerkverbindung kann jedoch erforderlich werden, wenn ein geöffnetes Dokument einen **externen Weblink** enthält und dieser vom Benutzer aufgerufen wird.

Der lokale Charakter der Dokumentanzeige ist deshalb von externen Links innerhalb der gelesenen Inhalte zu unterscheiden.

Bedeutung für JUMBO BLOCK Rescue

Reader stellt innerhalb von JUMBO BLOCK Rescue einen **lokalen universellen Dokumentenleser** bereit.

Damit können während eines Rescue-Einsatzes beispielsweise:

- technische Handbücher,
- Wartungsunterlagen,
- PDF-Berichte,
- E-Books,
- Markdown-Dokumentationen,
- Textdateien,
- lokale HTML-Dokumente,
- JSON-Daten,
- XML-Austauschdateien

direkt innerhalb der Rescue-Umgebung geöffnet werden.

Besonders JSON und XML ermöglichen dabei nicht nur das Lesen klassischer Dokumente, sondern auch die visuelle Prüfung strukturierter technischer Dateien.

Da die Dokumentverarbeitung lokal erfolgt, ist für das eigentliche Lesen keine Cloud- oder Serverinfrastruktur erforderlich.

Grenzen und Hinweise

Die Darstellung eines EPUB hängt auch von der Qualität und Struktur der jeweiligen EPUB-Datei ab.

Fehlerhafte oder ungewöhnlich aufgebaute E-Books können deshalb zu Einschränkungen führen.

Die PDF-Textsuche funktioniert ausschließlich bei PDF-Dateien mit auslesbarer Textebene.

Reader besitzt **keine OCR-Funktion**.

Externe Links innerhalb von EPUB- oder HTML-Dokumenten können eine Internetverbindung benötigen.

Der echte Browser-Vollbildmodus ist von Browser und Mobilgerät abhängig. Der Reader-Fokusmodus steht unabhängig davon zur Verfügung.

Browserbereinigung oder private Sitzungen können die dauerhafte Speicherung von:

- Lesepositionen,
- Lesezeichen,
- Markierungen,
- zuletzt gelesener Datei

verhindern beziehungsweise vorhandene Daten löschen.

Sehr große PDF- oder EPUB-Dokumente sowie Dokumente mit umfangreichen eingebetteten Bildern können abhängig von Gerät und verfügbarem Arbeitsspeicher längere Ladezeiten verursachen.

Technische Einordnung in JUMBO BLOCK Rescue

Reader ergänzt die lokalen Dokumentenwerkzeuge von JUMBO BLOCK Rescue um eine **reine Lese- und Navigationsanwendung**.

Die funktionale Zuordnung lautet:

Writer → Dokumente erstellen und bearbeiten

Reader → Dokumente, technische Unterlagen und E-Books lesen

Reader ist damit besonders für Situationen geeignet, in denen vorhandene Dokumentationen während eines IT-Ausfalls unmittelbar verfügbar sein müssen, ohne dass eine vollständige Office- oder Cloud-Infrastruktur bereitsteht.

Die Unterstützung von EPUB, PDF, Markdown, TXT, HTML, JSON und XML deckt dabei sowohl klassische Dokumente als auch technische Text- und Datenformate ab.

Lesezeichen, Markierungen, gespeicherte Lesepositionen sowie Fokus- und Vollbildmodus unterstützen zusätzlich die Nutzung umfangreicher Dokumentationen direkt innerhalb der Rescue-Umgebung.

Weiterführende Dokumentation

JUMBO BLOCK Reader – Benutzerhandbuch · Version 1.0.4

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Reader“ auf **https://jumboblock.org/.**

D.15 – Text

Bereich: Text / Code
Programm: JUMBO BLOCK Text
Baseline 20261001: 1.0.0 ff.

Zweck

JUMBO BLOCK Text ist der vollständig lokal arbeitende Text- und Code-Editor der JUMBO BLOCK Engineering Suite.

Das Programm dient zur Bearbeitung von **reinen Textdateien und Quelltexten** und ist damit bewusst von der formatierten Textverarbeitung JUMBO BLOCK Writer getrennt.

Text eignet sich insbesondere für:

- einfache Textdateien,
- Quellcode,
- Konfigurationsdateien,
- strukturierte Textformate,
- technische Notizen,
- Protokoll- und Logdateien,
- HTML-, CSS- und JavaScript-Dateien,
- JSON- und XML-Dateien.

Die Bearbeitung erfolgt vollständig lokal im Browser.

Rolle innerhalb der Engineering Suite

Text übernimmt die Funktion eines **allgemeinen lokalen Plain-Text- und Code-Editors**.

Die funktionale Abgrenzung lautet:

Writer → formatierte Dokumente und Textverarbeitung

Note → freie Notizen

Reader → Dokumente lesen und navigieren

Text → reine Textdateien und Quellcode öffnen, bearbeiten und speichern

Text arbeitet nicht mit der gemeinsamen `project.dbm` der projektspezifischen Engineering-Anwendungen.

Die jeweils geöffnete Datei bleibt die eigentliche Arbeitsdatei.

Wesentliche Funktionen

Dateien öffnen

Lokale Textdateien können über die Dateiauswahl geöffnet und in den Editor geladen werden.

Text arbeitet mit verschiedenen textbasierten Dateitypen und erkennt unterstützte Formate anhand der Datei beziehungsweise ihrer Endung.

Nach dem Öffnen steht der vollständige Textinhalt unmittelbar zur Bearbeitung zur Verfügung.

Neue Datei

Neben dem Öffnen vorhandener Dateien kann eine neue leere Textdatei angelegt werden.

Damit lassen sich auch ohne vorhandene Ausgangsdatei:

- Notizen,
- Konfigurationen,
- Quelltexte,
- strukturierte Daten,
- einfache Dokumentationen

erstellen.

Bearbeiten

Der Editor arbeitet als klassischer Plain-Text-Editor.

Inhalte werden als Zeichen und Zeilen bearbeitet. Anders als bei Writer existieren keine:

- Schriftformatierungen,
- Absatzformate,
- Seitenlayouts,
- eingebetteten Bilder.

Dadurch bleibt der eigentliche Textinhalt unmittelbar sichtbar und bearbeitbar.

Dies ist insbesondere bei Quellcode, Konfigurationsdateien und strukturierten Textformaten wichtig.

Syntaxhervorhebung

Text unterstützt Syntaxhervorhebung für verschiedene textbasierte Formate und Programmiersprachen.

Dadurch können beispielsweise:

- Schlüsselwörter,
- Zeichenketten,
- Kommentare,
- Zahlen,
- strukturelle Elemente

optisch voneinander unterschieden werden.

Die Syntaxhervorhebung dient ausschließlich der Darstellung im Editor.

Sie verändert den Inhalt der gespeicherten Datei nicht.

Damit eignet sich Text auch zur lokalen Prüfung und Bearbeitung von Quell- und Konfigurationsdateien innerhalb der Rescue-Umgebung.

Zeilennummern

Der Editor stellt Zeilennummern zur Verfügung.

Dadurch können bestimmte Positionen in:

- Quellcode,
- Konfigurationsdateien,
- Logdateien,
- technischen Textdateien

eindeutig referenziert werden.

Dies erleichtert insbesondere die Fehlersuche und die Abstimmung über konkrete Textstellen.

Suchen und Ersetzen

Text besitzt eine integrierte Suchfunktion.

Damit können Begriffe innerhalb der geöffneten Datei gezielt gefunden werden.

Zusätzlich steht eine Ersetzen-Funktion zur Verfügung, mit der vorhandene Textstellen verändert werden können.

Gerade bei umfangreichen:

- Konfigurationsdateien,
- Quelltexten,
- JSON-/XML-Dateien,
- Protokollen

können bestimmte Einträge dadurch schneller lokalisiert und bearbeitet werden.

Rückgängig und Wiederholen

Bearbeitungsschritte können zurückgenommen und erneut ausgeführt werden.

Damit lassen sich versehentliche Änderungen während der laufenden Bearbeitung korrigieren, solange der entsprechende Bearbeitungszustand noch in der geöffneten Sitzung verfügbar ist.

Daten und Dateiformate

Text arbeitet unmittelbar mit textbasierten Dateien.

Es verwendet kein eigenes Projektformat und keine gemeinsame `project.dbm`.

Damit unterscheidet sich Text von Anwendungen wie:

- Project,
- Note,
- Tables,
- Writer,

die eigene strukturierte Arbeitsdateien beziehungsweise Datenmodelle besitzen.

Text bleibt bewusst auf den Inhalt der jeweils bearbeiteten Textdatei ausgerichtet.

Dies ermöglicht insbesondere die Bearbeitung von Dateien, die auch mit anderen Texteditoren weiterverwendet werden können.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Text läuft vollständig lokal im Browser.

Für die eigentliche Textbearbeitung werden keine Cloud- oder Serverdienste benötigt.

Damit steht innerhalb von JUMBO BLOCK Rescue ein allgemeiner Text- und Code-Editor zur Verfügung, der unabhängig von einer installierten Entwicklungsumgebung oder einer Online-Anwendung eingesetzt werden kann.

Das ist im Rescue-Kontext insbesondere relevant für die Bearbeitung beziehungsweise Prüfung von:

- Konfigurationsdateien,
- Skripten,
- HTML-Dateien,
- CSS-Dateien,
- JavaScript-Dateien,
- JSON-Dateien,

- XML-Dateien,
- Logdateien,
- sonstigen reinen Textdateien.

Text ergänzt damit sowohl die technischen Rescue-Werkzeuge als auch die Engineering Suite um eine unmittelbar verfügbare lokale Bearbeitungsmöglichkeit für textbasierte Daten.

Datensicherung

Da Text mit den eigentlichen Textdateien arbeitet, muss ein bearbeiteter Stand bewusst als Datei gespeichert werden.

Vor Änderungen an wichtigen:

- Systemdateien,
- Konfigurationsdateien,
- Quelltexten,
- Projektdateien in lesbarer Textform

sollte grundsätzlich eine unveränderte Sicherungskopie des Ausgangsstands angelegt werden.

Dies ist innerhalb einer Rescue-Umgebung besonders wichtig.

Eine syntaktisch kleine Änderung kann bei Konfigurations- oder Programmdateien erhebliche Auswirkungen auf das zugehörige System haben.

Sprache

Text ist in die deutsch-/englischsprachige Benutzeroberfläche der JUMBO BLOCK Engineering Suite eingebunden.

Die Sprachumschaltung betrifft die Bedienoberfläche.

Der Inhalt der geöffneten Textdatei wird dadurch nicht verändert oder automatisch übersetzt.

Damit können auch Dateien in anderen Sprachen oder Programmiersprachen unabhängig von der gewählten Oberflächensprache bearbeitet werden.

Grenzen und Hinweise

Text ist ein **Plain-Text- und Code-Editor**.

Er ist keine formatierte Textverarbeitung.

Für Dokumente mit:

- Seitenlayout,
- Schriftformatierung,
- Tabellenlayout,

- eingebetteten Bildern,
- Briefkopf,
- Inhaltsverzeichnis

ist JUMBO BLOCK Writer vorgesehen.

Text ist außerdem keine vollständige integrierte Entwicklungsumgebung.

Die Syntaxhervorhebung unterstützt die Lesbarkeit von Quelltext, führt jedoch keine fachliche Prüfung des Programmcodes durch.

Bei Änderungen an System- und Konfigurationsdateien ist besondere Vorsicht erforderlich.

Das erfolgreiche Speichern einer Datei bedeutet nicht automatisch, dass deren Inhalt:

- syntaktisch korrekt,
- fachlich korrekt,
- mit dem Zielsystem kompatibel

ist.

Vor Eingriffen in wichtige Dateien sollte deshalb der ursprüngliche Datenstand gesichert werden.

Technische Einordnung in JUMBO BLOCK Rescue

Text ergänzt JUMBO BLOCK Rescue um einen **allgemeinen lokalen Editor für reine Text- und Codedateien**.

Die funktionale Zuordnung lautet:

Note → schnelle freie Notizen

Writer → formatierte Dokumente

Reader → Dokumente lesen

Text → Text-, Konfigurations- und Quellcodedateien bearbeiten

Damit besitzt JUMBO BLOCK Rescue neben seinen Diagnose-, Datenrettungs-, Engineering- und Dokumentenwerkzeugen auch eine unmittelbar verfügbare Möglichkeit, textbasierte Dateien direkt zu untersuchen und bei Bedarf zu bearbeiten.

Gerade bei Systemrettung und Wiederherstellung ist diese Funktion relevant, weil zahlreiche technische Informationen und Konfigurationen in reinen Textformaten vorliegen.

Die vollständig lokale Arbeitsweise ermöglicht diese Bearbeitung auch dann, wenn keine reguläre Entwicklungs-, Office- oder Cloud-Infrastruktur verfügbar ist.

Weiterführende Dokumentation

JUMBO BLOCK Text – Benutzerhandbuch · Version 1.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Text“ auf **https://jumboblock.org/.**

D.16 – PDF Merge

Bereich: PDF
Programm: JUMBO BLOCK PDF Merge
Baseline 20261001: 1.0.0 ff.

Zweck

JUMBO BLOCK PDF Merge ist ein bewusst klein und einfach gehaltenes Werkzeug der JUMBO BLOCK Engineering Suite zum:

- Zusammenführen mehrerer PDF-Dateien zu einer neuen PDF-Datei,
- Ändern der Seitenreihenfolge,
- Entfernen einzelner Seiten aus einer PDF-Datei.

Beide Kernfunktionen – **Zusammenführen und Seiten entfernen** – basieren auf einer gemeinsamen geordneten Seitenliste.

Werden mehrere PDF-Dateien geladen, erscheinen deren Seiten gemeinsam in dieser Liste. Wird nur eine PDF-Datei geladen und werden anschließend einzelne Seiten entfernt, entsteht daraus beim Speichern eine reduzierte Version der ursprünglichen Datei.

Die Verarbeitung erfolgt vollständig lokal.

Rolle innerhalb der Engineering Suite

PDF Merge übernimmt innerhalb der Engineering Suite eine klar abgegrenzte Aufgabe:

vorhandene PDF-Seiten auswählen, neu anordnen und zu einer neuen PDF-Datei zusammenführen.

Das Programm bearbeitet nicht den eigentlichen Inhalt einer PDF-Seite.

Die funktionale Abgrenzung lautet:

Writer → Dokumente erstellen und als PDF ausgeben

Reader → vorhandene PDF-Dokumente lesen

PDF Merge → vorhandene PDF-Seiten auswählen, sortieren und zusammenführen

PDF Printer → Dokumente beziehungsweise unterstützte Inhalte als PDF ausgeben

PDF Merge arbeitet nicht mit der gemeinsamen `project.dbm` der projektspezifischen Engineering-Anwendungen.

PDFs hinzufügen

PDF-Dateien können auf zwei Wegen hinzugefügt werden.

Dateiauswahl

Über:

„ PDF(s) hinzufügen“

wird der Datei-Auswahldialog des Betriebssystems geöffnet.

Eine Mehrfachauswahl ist möglich.

Drag-and-drop

PDF-Dateien können direkt auf die dafür vorgesehene Ablagefläche gezogen werden.

Ein Klick auf diese Fläche öffnet ebenfalls die Dateiauswahl.

Die Ablagefläche ist zusätzlich per Tastatur bedienbar:

- Tabulatortaste zum Fokussieren,
- Eingabetaste oder Leertaste zum Öffnen der Dateiauswahl.

Laden der PDF-Dateien

Jede ausgewählte Datei wird einzeln geladen.

Währenddessen zeigt PDF Merge den jeweiligen Dateinamen im Status an.

Ist eine Datei:

- kein gültiges PDF,
- verschlüsselt,

wird sie abgewiesen.

Andere gleichzeitig ausgewählte gültige Dateien werden davon unabhängig weiterverarbeitet.

Damit führt eine einzelne ungeeignete Datei nicht zum Abbruch des gesamten Ladevorgangs.

Gemeinsame Seitenliste

Jede erfolgreich geladene PDF-Seite erscheint als eigene Zeile in der Seitenliste.

Angezeigt werden:

- Name der Quelldatei,
- ursprüngliche Seitennummer,
- Gesamtseitenzahl der Quelldatei.

Die Darstellung folgt beispielsweise dem Schema:

„Dateiname – Seite X von Y“

Die Seiten unterschiedlicher Quelldateien werden am linken Rand farblich unterschieden.

Dafür verwendet PDF Merge eine Palette aus sechs Farben. Werden mehr als sechs PDF-Dateien geladen, wiederholt sich diese Farbpalette.

Über der Seitenliste wird zusätzlich zusammengefasst, wie viele Seiten aus wie vielen Dateien geladen wurden.

Entscheidend ist:

Die Reihenfolge dieser Liste entspricht exakt der späteren Seitenreihenfolge der erzeugten PDF-Datei.

Seiten anordnen

Für jede Seite stehen mehrere Möglichkeiten zur Verfügung.

Eine Position verschieben

Mit:

▲

wird die Seite um eine Position nach oben verschoben.

Mit:

▼

wird sie um eine Position nach unten verschoben.

Am oberen beziehungsweise unteren Ende der Liste ist die jeweils nicht mögliche Funktion deaktiviert.

Freies Verschieben

Eine Seite kann außerdem mit der Maus direkt an eine andere Position gezogen werden.

Während des Ziehens zeigt eine Linie, ob die Seite oberhalb oder unterhalb der jeweiligen Zielzeile eingefügt wird.

Damit können auch umfangreichere Seitenfolgen unmittelbar neu aufgebaut werden.

Seiten entfernen

Über:

×

wird die jeweilige Seite aus der Arbeitsliste entfernt.

Dabei wird ausschließlich der Eintrag in PDF Merge entfernt.

Die ursprüngliche PDF-Datei wird nicht verändert.

Auf diese Weise können beispielsweise einzelne nicht benötigte Seiten aus einer PDF entfernt werden, indem die verbleibenden Seiten anschließend als neue PDF-Datei gespeichert werden.

Speichern

Beim ersten Hinzufügen von Seiten erzeugt PDF Merge automatisch einen vorgeschlagenen Dateinamen nach dem Schema:

yyyymmdd_hhmmss_pdfmerge.pdf

Beispiel:

20260811_093000_pdfmerge.pdf

Der Dateiname kann vor dem Speichern frei geändert werden.

Fehlt die Dateiendung:

.pdf

wird sie automatisch ergänzt.

Über:

„ Speichern“

erzeugt PDF Merge aus der aktuellen Seitenliste eine neue PDF-Datei.

Die Seiten werden dabei exakt in der angezeigten Reihenfolge übernommen.

Ist die Seitenliste leer, wird keine Datei erzeugt.

Speicherort und Browserunterschiede

Das Verhalten beim Speichern hängt teilweise vom verwendeten Browser ab.

Chrome und Edge

In Chrome und Edge kann der native Betriebssystemdialog:

„Speichern unter“

geöffnet werden.

Dort können:

- Zielordner,
- Dateiname

gewählt werden.

Der zuvor in PDF Merge eingetragene Dateiname wird bereits vorgeschlagen.

Wird der Dialog abgebrochen, erzeugt PDF Merge keine Datei.

Firefox und Safari

Firefox und Safari stellen diese freie Ordnerauswahl für das betreffende Browser-Speicherverfahren grundsätzlich nicht zur Verfügung.

PDF Merge weist in diesen Browsern darauf hin.

Die erzeugte Datei wird dann unter dem angegebenen Dateinamen im Standard-Download-Ordner des Browsers gespeichert.

Diese Einschränkung ergibt sich aus dem jeweiligen Browser und nicht aus der PDF-Verarbeitung von PDF Merge.

Datenintegrität

Nach jedem erfolgreichen Speichern berechnet PDF Merge die:

SHA-256-Prüfsumme

der neu erzeugten PDF-Datei.

In der Statuszeile werden die ersten **16 Zeichen** des SHA-256-Wertes angezeigt.

Damit kann später geprüft werden, ob eine vorhandene Datei exakt dem Datenstand entspricht, den PDF Merge zum Zeitpunkt des Speicherns erzeugt hat.

Der Hash bezieht sich auf die erzeugte PDF-Datei selbst.

Zurücksetzen

Über:

„♻ Zurücksetzen“

kann der aktuelle Arbeitsstand vollständig geleert werden.

Dabei werden:

- Seitenliste,
- geladene Quelldateien im Arbeitsspeicher,
- Dateinamensfeld

zurückgesetzt.

Sind bereits Dateien oder Seiten geladen, erscheint zuvor eine Sicherheitsabfrage.

Erst nach deren Bestätigung wird der aktuelle Arbeitsstand verworfen.

Die ursprünglichen PDF-Dateien auf dem Datenträger werden dadurch nicht verändert.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

PDF Merge läuft vollständig lokal im Browser.

Es gibt:

- keine Cloud-Verarbeitung,
- keinen Upload auf einen Server,
- kein Benutzerkonto.

Sämtliche PDF-Dateien werden auf dem Gerät verarbeitet, auf dem PDF Merge ausgeführt wird.

Auch die neue PDF-Datei wird lokal erzeugt.

Damit können innerhalb von JUMBO BLOCK Rescue beispielsweise:

- gerettete PDF-Dokumente zusammengeführt,
- Berichte in eine gewünschte Reihenfolge gebracht,
- einzelne unerwünschte Seiten entfernt,
- mehrere technische Unterlagen zu einer gemeinsamen Datei kombiniert

werden, ohne die Dokumente an einen externen Dienst übertragen zu müssen.

Sitzungsbezogene Arbeitsweise

PDF Merge speichert den aufgebauten Arbeitsstand **nicht zwischen Sitzungen**.

Jede neue Sitzung beginnt mit einer leeren Seitenliste.

Nach:

- Neuladen der Seite,
- Schließen des Browser-Tabs

sind die geladenen Dateien und die aufgebaute Seitenreihenfolge nicht mehr vorhanden.

Ein benötigtes Ergebnis muss deshalb vor dem Beenden der Sitzung als neue PDF-Datei gespeichert werden.

PDF Merge besitzt kein eigenes Projekt- oder Sitzungsdateiformat zum späteren Fortsetzen einer begonnenen Zusammenstellung.

Sprache

PDF Merge unterstützt Deutsch und Englisch.

Die Umschaltung erfolgt über das gemeinsame Menü der JUMBO BLOCK Engineering Suite.

Dabei werden:

- Beschriftungen,
- Statusmeldungen,
- Hinweistexte

entsprechend umgeschaltet.

Die geladenen PDF-Inhalte selbst werden dadurch selbstverständlich nicht verändert.

Grenzen und Hinweise

PDF Merge ist bewusst auf das **Anordnen und Auswählen vorhandener PDF-Seiten** beschränkt.

Keine Seitenvorschau

Die Seiten werden als beschriftete Zeilen dargestellt.

Es werden keine Miniaturansichten der einzelnen PDF-Seiten erzeugt.

Das Programm bleibt dadurch bewusst klein und benötigt keine zusätzliche umfangreiche Bibliothek für die Bildvorschau.

Keine Bearbeitung von Seiteninhalten

PDF Merge kann keine Inhalte innerhalb einer Seite verändern.

Insbesondere stehen keine Funktionen zur Verfügung für:

- Drehen von Seiten,
- Einfügen von Wasserzeichen,
- Bearbeiten von Text,
- Bearbeiten von Bildern oder sonstigen Seiteninhalten.

Verändert werden ausschließlich:

- Auswahl der Seiten,
- Reihenfolge der Seiten.

Keine verschlüsselten PDF-Dateien

Passwortgeschützte beziehungsweise verschlüsselte PDF-Dateien werden beim Laden abgewiesen.

Keine dauerhafte Sitzung

Geladene Dateien und die Seitenreihenfolge werden nicht zwischen Sitzungen gespeichert.

Browserabhängiger Speicherort

Die freie Wahl des Zielordners ist vom verwendeten Browser abhängig.

Unter Firefox und Safari wird die Datei über den Standard-Download-Mechanismus des Browsers gespeichert.

Technische Einordnung in JUMBO BLOCK Rescue

PDF Merge ergänzt JUMBO BLOCK Rescue um ein **kleines, vollständig lokales Werkzeug zur strukturellen Neuordnung vorhandener PDF-Dokumente**.

Die funktionale Zuordnung lautet:

Reader → PDF-Dateien lesen

Writer → Dokumente erstellen und PDF ausgeben

PDF Merge → PDF-Seiten zusammenführen, sortieren und entfernen

PDF Printer → Inhalte in eine PDF-Ausgabe überführen

Besonders im Rescue-Einsatz ist die lokale Verarbeitung relevant, weil wiederhergestellte oder vertrauliche Dokumente nicht für eine einfache PDF-Zusammenführung an einen externen Online-Dienst übertragen werden müssen.

Die ursprünglichen Dateien bleiben dabei unverändert. Erst beim Speichern wird aus der aktuellen Seitenliste eine neue PDF-Datei erzeugt.

Der anschließend berechnete SHA-256-Wert ermöglicht zusätzlich die Überprüfung, ob diese erzeugte Ausgabedatei später unverändert vorliegt.

Weiterführende Dokumentation

JUMBO BLOCK PDF Merge – Benutzerhandbuch · Version 1.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „PDF Merge“ auf **https://jumboblock.org/.**

D.17 – PDF Printer

Bereich: PDF
Programm: JUMBO BLOCK PDF Printer
Baseline 20261001: 1.2.1 ff.

Zweck

JUMBO BLOCK PDF Printer ist ein browserbasiertes, lokal arbeitendes Werkzeug zur:

- Anzeige,
- Prüfung,
- Druckvorbereitung,
- direkten Ausgabe

vorhandener PDF-Dokumente über den angeschlossenen Systemdrucker.

Die PDF-Datei wird vollständig lokal im Browser verarbeitet. Dokumente werden nicht an einen externen Server übertragen.

Neben der normalen Seitenausgabe unterstützt PDF Printer insbesondere:

- unterschiedliche Papierformate und Ausrichtungen,
- mehrere PDF-Seiten auf einem Druckblatt,
- Broschürendruck,
- unterschiedliche Renderqualitäten,
- eine interaktive Seitenvorschau.

Damit dient PDF Printer als spezialisierte lokale Druckoberfläche für PDF-Dokumente.

Rolle innerhalb der Engineering Suite

PDF Printer übernimmt innerhalb der Engineering Suite die **lokale Druckaufbereitung vorhandener PDF-Dateien**.

Die funktionale Abgrenzung lautet:

Writer → Dokumente erstellen und als PDF ausgeben

Reader → PDF-Dokumente lesen

PDF Merge → vorhandene PDF-Seiten zusammenführen, sortieren und entfernen

PDF Printer → vorhandene PDF-Dokumente prüfen, für den Druck aufbereiten und über den Systemdrucker ausgeben

PDF Printer verändert dabei nicht den eigentlichen Inhalt des Quelldokuments.

Es arbeitet nicht mit der gemeinsamen `project.dbm` der projektspezifischen Engineering-Anwendungen.

PDF öffnen

Eine PDF-Datei kann über die lokale Dateiauswahl geöffnet werden.

Alternativ unterstützt PDF Printer:

Drag & Drop

direkt in den Vorschaubereich.

Die Verarbeitung findet anschließend innerhalb des Browsers statt.

Interaktive Vorschau

Nach dem Öffnen wird das PDF in einer interaktiven Vorschau dargestellt.

Eine Seitenleiste enthält Miniaturansichten der einzelnen PDF-Seiten.

Dadurch kann das Dokument vor dem eigentlichen Ausdruck visuell kontrolliert und gezielt navigiert werden.

Die aktive Seite kann ausgewählt werden über:

- die Schaltflächen **◀ / ▶**,
- einen Klick auf die entsprechende Miniaturansicht.

Die Vorschau unterscheidet PDF Printer damit deutlich vom bewusst reduzierten PDF Merge, das keine Seiten-Thumbnails erzeugt.

Zoom und Seitenanpassung

Für die Dokumentvorschau stehen mehrere Zoomfunktionen zur Verfügung.

Vergrößern und Verkleinern

Über:

+

beziehungsweise:

–

kann die Darstellung vergrößert oder verkleinert werden.

100 %

Die Funktion:

100 %

setzt die Darstellung auf die entsprechende Zoomstufe zurück.

Breite

„Breite“

passt die PDF-Seite an die verfügbare Breite des Vorschaubereichs an.

Seite

„Seite“

stellt die vollständige PDF-Seite innerhalb des verfügbaren Vorschaubereichs dar.

Stufenloser Zoom

Zusätzlich kann mit:

Strg + Mausrad

stufenlos innerhalb der Vorschau gezoomt werden.

Papierformat

Für die Druckausgabe stehen laut Handbuch folgende Papierformate zur Verfügung:

- A4,
- A3,
- Letter.

Damit kann das Zielpapier unabhängig vom ursprünglichen PDF-Seitenformat für die Druckaufbereitung festgelegt werden.

Seitenausrichtung

Für das gewählte Zielpapier kann zwischen:

- Hochformat,
- Querformat

gewählt werden.

Papierformat und Orientierung bilden gemeinsam die Grundlage für die nachfolgende Druckaufbereitung.

Druckqualität

PDF Printer stellt drei Qualitätsstufen für das Rendering zur Verfügung.

Entwurf

Schnelles Rendering mit geringerer Auflösung.

Diese Einstellung eignet sich insbesondere für schnelle Kontrollausdrucke.

Gut

Standard-Rendering für allgemeine Dokumente.

Hoch

Maximale Renderqualität.

Diese Einstellung ist insbesondere für:

- Pläne,
- Bilder,
- Präsentationen

vorgesehen.

Die Qualitätswahl beeinflusst damit die Aufbereitung der PDF-Seiten für die Druckausgabe.

Seiten pro Blatt – N-up

PDF Printer kann mehrere PDF-Seiten auf einem einzelnen Druckblatt anordnen.

Unterstützt werden:

- 1 Seite pro Blatt,
- 2 Seiten pro Blatt,
- 4 Seiten pro Blatt.

Diese sogenannte **N-up-Ausgabe** ermöglicht beispielsweise:

- platzsparende Kontrollausdrucke,
- kompakte Dokumentationen,
- verkleinerte Mehrseitenausgaben.

Die Seiten werden entsprechend der gewählten Einstellung auf dem Zielpapier angeordnet.

Broschürendruck

Eine besondere Funktion von PDF Printer ist:

„Als Broschüre drucken“

Im Broschürenmodus ordnet das Programm die Seiten automatisch so um, dass nach:

- zweiseitigem Druck,
- Falten,
- Heften

ein fertiges Heft entstehen kann.

Die für den Broschürendruck erforderliche Seitenreihenfolge wird automatisch erzeugt.

Bei aktiviertem Broschürenmodus wird die normale N-up-Auswahl deaktiviert.

Duplex-Hinweis

PDF Printer unterstützt die Vorbereitung für beidseitigen Druck durch entsprechende Hinweise.

Insbesondere beim Broschürendruck gibt das Programm Empfehlungen für die erforderliche Duplex-Einstellung.

Ein Beispiel ist:

Bindung an der kurzen Kante

für eine entsprechende Broschürenausgabe.

Die eigentliche Duplex-Funktion wird vom angeschlossenen Drucker beziehungsweise dessen System-Druckdialog ausgeführt.

PDF Printer bereitet die Seitenanordnung vor und weist auf die passende Druckeinstellung hin.

Druckausgabe

Die endgültige Ausgabe erfolgt über den angeschlossenen **Systemdrucker**.

Über den Druckdialog des Systems beziehungsweise Browsers können die tatsächlich vom jeweiligen Drucker bereitgestellten Optionen verwendet werden.

Dazu können abhängig von:

- Betriebssystem,
- Browser,
- Druckertreiber,
- Druckermodell

weitere Einstellungen gehören.

PDF Printer selbst konzentriert sich auf die vorherige Aufbereitung und Anordnung der PDF-Seiten.

Tastaturbedienung

PDF Printer unterstützt mehrere Tastenkombinationen.

Seitennavigation

Pfeil links
vorherige Seite

Pfeil rechts
nächste Seite

Zoom

+ beziehungsweise **=**
einzoomen

–
auszoomen

Strg + Mausrad
stufenloser Zoom im Vorschaubereich

An Breite anpassen

F

passt die Darstellung an die verfügbare Breite an.

100 %

0

setzt den Zoom auf 100 % zurück.

Drucken

Strg + P

öffnet unmittelbar den Druckdialog.

Damit kann PDF Printer auch bei häufig wiederkehrenden Prüf- und Druckaufgaben weitgehend über die Tastatur bedient werden.

Daten und Datenaustausch

PDF Printer verwendet kein eigenes Projekt- oder Dokumentformat.

Die jeweils geöffnete PDF-Datei bleibt das Ausgangsdokument.

Das Programm dient der:

- Anzeige,
- Prüfung,
- Druckaufbereitung,
- Ausgabe

dieser Datei.

Es findet kein Datenaustausch mit der gemeinsamen `project.dbm` der Engineering Suite statt.

Auch ein eigenes PDF-Printer-Projektformat ist im zugrunde liegenden Handbuch nicht dokumentiert.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

PDF Printer verarbeitet die geöffneten PDF-Dokumente vollständig lokal im Browser.

Es findet kein Upload und keine Übertragung des Dokuments an einen externen Server statt.

Damit können innerhalb von JUMBO BLOCK Rescue beispielsweise:

- technische Berichte,
- gerettete PDF-Dokumente,
- Pläne,
- Handbücher,
- Projektunterlagen,
- Dokumentationen

lokal geprüft und für einen vorhandenen Drucker aufbereitet werden.

Gerade bei vertraulichen oder aus einem beschädigten System geretteten Dokumenten entfällt dadurch die Notwendigkeit, für die Druckaufbereitung einen externen Online-PDF-Dienst zu verwenden.

Bedeutung des Systemdruckers

PDF Printer ist kein eigener Druckertreiber.

Die endgültige Ausgabe erfolgt über die Druckfunktion des verwendeten Systems.

Damit hängt die tatsächlich mögliche physische Ausgabe auch davon ab, ob innerhalb der Rescue-Umgebung:

- der Drucker erkannt wird,
- ein geeigneter Druckertreiber beziehungsweise eine passende Systemunterstützung vorhanden ist,
- die gewünschte Duplex- oder Papierfunktion vom Drucker unterstützt wird.

PDF Printer übernimmt die browserseitige Dokumentdarstellung und Druckaufbereitung; die physische Druckausgabe erfolgt anschließend über die vorhandene System-Druckumgebung.

Datenschutz

Die lokale Verarbeitung ist ein zentrales Merkmal von PDF Printer.

Die geöffnete PDF-Datei wird für die Dokumentdarstellung nicht auf einen externen Server übertragen.

Dadurch können auch vertrauliche Unterlagen innerhalb der lokalen Rescue-Umgebung für den Druck vorbereitet werden.

Dies unterscheidet PDF Printer insbesondere von webbasierten Druck- oder PDF-Diensten, die Dokumente zunächst auf einen externen Server hochladen müssen.

Technische Grundlage

Für die PDF-Darstellung verwendet JUMBO BLOCK PDF Printer:

PDF.js

der Mozilla Foundation.

PDF.js steht unter der:

Apache License 2.0

Die eigenen:

- HTML-Bestandteile,
- CSS-Bestandteile,
- JavaScript-Programmlogik

des JUMBO BLOCK PDF Printer stammen von der ZANNI GROUP.

Grenzen und Hinweise

PDF Printer ist ein Werkzeug zur **Anzeige und Druckaufbereitung vorhandener PDF-Dokumente**.

Es ist kein PDF-Inhaltseditor.

Der eigentliche Seiteninhalt wird nicht bearbeitet.

Für das Zusammenführen, Entfernen oder Neuordnen einzelner PDF-Seiten ist **JUMBO BLOCK PDF Merge** vorgesehen.

Die tatsächliche Druckausgabe hängt von der vorhandenen System-Druckumgebung ab.

Insbesondere Funktionen wie:

- Duplexdruck,
- unterstützte Papierformate,
- Druckauflösung,
- Druckerfächer,
- Randverhalten

können vom verwendeten Drucker und dessen Systemunterstützung abhängig sein.

Der Broschürenmodus bereitet die Seitenreihenfolge vor. Für die korrekte physische Broschüre müssen zusätzlich die passenden Duplex-, Falt- und gegebenenfalls Heftbedingungen des jeweiligen Drucksystems eingehalten werden.

Die im zugrunde liegenden Handbuch dokumentierten Funktionen beziehen sich auf **Version 1.2.0**. Die eingefrorene JUMBO BLOCK Rescue Baseline 20261001 enthält **Version 1.2.1**. Änderungen zwischen diesen beiden Versionen werden hier nicht angenommen, solange sie nicht durch die vorliegende Programmdokumentation verifiziert sind.

Technische Einordnung in JUMBO BLOCK Rescue

PDF Printer ergänzt JUMBO BLOCK Rescue um eine **vollständig lokale PDF-Vorschau und spezialisierte Druckaufbereitung**.

Die funktionale Zuordnung lautet:

Reader → PDF-Dokumente lesen

PDF Merge → PDF-Seiten zusammenführen, sortieren und entfernen

PDF Printer → PDF-Dokumente für die physische Ausgabe aufbereiten und drucken

Mit:

- Seitenvorschau,
- Miniaturansichten,
- Zoom,
- A4/A3/Letter,
- Hoch- und Querformat,
- drei Renderqualitäten,
- 1/2/4 Seiten pro Blatt,
- Broschürendruck

deckt PDF Printer die wesentlichen im Handbuch dokumentierten Funktionen für die lokale Druckvorbereitung ab.

Damit können auch während eines IT-Ausfalls vorhandene PDF-Unterlagen innerhalb der Rescue-Umgebung geprüft und für einen verfügbaren Systemdrucker vorbereitet werden, ohne die Dokumente an externe Dienste zu übertragen.

Weiterführende Dokumentation

JUMBO BLOCK PDF Printer – Benutzerhandbuch · verfügbare Dokumentation Version 1.2.0

JUMBO BLOCK Rescue Baseline 20261001: PDF Printer 1.2.1 ff.

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „PDF Printer“ auf **https://jumboblock.org/.**

D.18 – Convert

Bereich: Konvertierung
Programm: JUMBO BLOCK Convert
Baseline 20261001: 0.3.2 ff.

Zweck

JUMBO BLOCK Convert ist das lokale Konvertierungswerkzeug der JUMBO BLOCK Engineering Suite für den Weg:

PDF → Word (.docx)

Viele Programme der Engineering Suite erzeugen ihre Berichte bewusst ausschließlich als PDF. Convert schließt die Lücke, wenn der Inhalt eines solchen PDF-Dokuments anschließend in **Microsoft Word oder einer kompatiblen Textverarbeitung weiterbearbeitet** werden soll.

Das Ziel der Konvertierung ist ausdrücklich **keine pixelgenaue optische 1:1-Kopie** des ursprünglichen PDF-Dokuments.

Priorität besitzt stattdessen:

ein möglichst gut bearbeitbares Word-Dokument mit sauber strukturiertem Fließtext.

Kleinere optische Abweichungen gegenüber der PDF-Vorlage werden dafür bewusst in Kauf genommen.

Rolle innerhalb der Engineering Suite

Convert bildet eine Brücke zwischen den PDF-Ausgaben der Engineering Suite und einer anschließenden editierbaren Word-Bearbeitung.

Die funktionale Abgrenzung lautet:

Writer → Dokumente erstellen und bearbeiten

Reader → vorhandene Dokumente lesen

PDF Merge → PDF-Seiten zusammenführen und neu anordnen

PDF Printer → PDF-Dokumente für den Druck aufbereiten

Convert → vorhandene PDF-Inhalte in ein bearbeitbares Word-Dokument überführen

Convert arbeitet nicht mit der gemeinsamen `project.dbm` der projektspezifischen Engineering-Anwendungen.

Ein eigenes Projekt- oder Arbeitsformat ist ebenfalls nicht vorgesehen.

Die Eingabedatei ist eine vorhandene PDF-Datei; das Ergebnis ist eine neue `.docx`-Datei.

Bedienung

Die Bedienung ist bewusst auf drei Arbeitsschritte reduziert.

1. PDF auswählen

Über:

„PDF auswählen“

wird die lokale Dateiauswahl des Betriebssystems geöffnet.

Akzeptiert werden ausschließlich PDF-Dateien.

2. Konvertieren

Nach Auswahl einer gültigen PDF-Datei wird:

„Konvertieren“

aktiv.

Ein Klick startet die vollständige Umwandlung.

Es gibt:

- keine Dokumentvorschau,
- keine zusätzlichen Konvertierungseinstellungen.

Die Verarbeitung läuft automatisch ab.

3. Word-Datei speichern

Nach erfolgreicher Konvertierung wird die erzeugte `.docx`-Datei gespeichert beziehungsweise über den Download-Mechanismus des Browsers ausgegeben.

Statusanzeige

Unterhalb der Schaltflächen informiert ein Statustext über den aktuellen Zustand.

Mögliche Meldungen sind:

- **Keine PDF ausgewählt**
- **Wird konvertiert …**
- **Fertig**
- **Abgebrochen**
- **Konvertierung fehlgeschlagen**

Damit ist jederzeit erkennbar, ob:

- noch keine Datei ausgewählt wurde,
- die Verarbeitung läuft,

- die Konvertierung erfolgreich abgeschlossen wurde,
- der Speichervorgang abgebrochen wurde,
- während der Verarbeitung ein Fehler aufgetreten ist.

Wie die Umwandlung funktioniert

Beim Start der Konvertierung durchläuft die PDF-Datei mehrere automatische Verarbeitungsschritte.

Textextraktion

Für jede PDF-Seite werden die vorhandenen Textzeichen einschließlich Informationen zu:

- Position,
- Schriftart,
- Schriftgröße

ausgelesen.

Diese Textschicht bildet die Grundlage für die anschließende Rekonstruktion des Word-Dokuments.

Kopf- und Fußzeilenerkennung

Convert versucht wiederkehrende Kopf- und Fußzeilen automatisch zu erkennen.

Textzeilen werden als entsprechende Word-Kopf- beziehungsweise -Fußzeile behandelt, wenn sie:

- identisch auf mindestens zwei Seiten im oberen beziehungsweise unteren Randbereich wiederkehren

oder

- einem typischen Seitenzahl-Muster entsprechen.

Ein Beispiel ist:

„Seite 3“

Erkannte Kopf- und Fußzeilen werden nicht einfach in den normalen Fließtext übernommen, sondern als echte Word-Kopf- beziehungsweise -Fußzeilen angelegt.

Absatzrekonstruktion

PDF-Dateien speichern Text häufig zeilen- oder positionsorientiert.

Eine einfache Übernahme jeder PDF-Zeile als einzelner Word-Absatz würde deshalb zu stark zerstückelten Dokumenten führen.

Convert versucht stattdessen, zusammengehörende Textzeilen anhand von:

- Zeilenabstand,
- Einzug,
- Schriftwechsel

zu zusammenhängenden Absätzen zu rekonstruieren.

Das Ziel ist ein **editierbarer Fließtext**, der sich anschließend in Word sinnvoll weiterbearbeiten lässt.

Tabellenerkennung

Convert versucht tabellarische Strukturen automatisch zu erkennen.

Als Tabelle werden Folgen von mindestens:

drei aufeinanderfolgenden Zeilen

erkannt, wenn deren Spaltenpositionen übereinstimmen.

Eine erkannte Struktur wird als **echte bearbeitbare Word-Tabelle** aufgebaut.

Damit bleiben geeignete tabellarische PDF-Inhalte nicht lediglich als optisch ausgerichteter Text erhalten.

Bilder

Eingebettete Rastergrafiken können aus der PDF-Datei extrahiert und anschließend in das Word-Dokument eingebettet werden.

Dies betrifft beispielsweise:

- Fotos,
- Logos,
- andere als Pixelbild gespeicherte Grafiken.

Nicht unterstützt werden dagegen PDF-Vektorgrafiken, die als Pfade und nicht als Rasterbilder gespeichert sind.

Schriftart-Zuordnung

PDF-Schriftnamen werden bei der Konvertierung auf möglichst passende Standard-Word-Schriften abgebildet.

Beispiele aus dem Handbuch sind:

Helvetica → Arial

Times → Times New Roman

Für unbekannte beziehungsweise eingebettete Schriftarten verwendet Convert als Rückfalllösung:

Calibri

Die Schriftzuordnung unterstützt das Ziel, ein anschließend gut bearbeitbares Word-Dokument zu erzeugen.

Eine exakte Reproduktion jeder im PDF verwendeten Schriftart ist dagegen nicht das Ziel.

Daten und Datenaustausch

Convert besitzt einen bewusst einfachen Datenfluss:

PDF-Datei → Convert → DOCX-Datei

Es gibt:

- kein eigenes Convert-Projektformat,
- keinen Austausch mit `project.dbm`,
- keine dokumentierte Zwischendatei.

Die PDF-Datei bleibt unverändert.

Das Ergebnis der Konvertierung wird als neue Word-Datei gespeichert.

Der vorgeschlagene Dateiname entspricht dem ursprünglichen Namen der PDF-Datei, wobei:

.pdf

durch:

.docx

ersetzt wird.

Speicherort wählen

Das Speicherverhalten hängt vom verwendeten Browser ab.

Chrome und Edge

Unterstützt der Browser die **File System Access API**, kann nach Abschluss der Konvertierung der native:

„Speichern unter“-Dialog

des Betriebssystems geöffnet werden.

Dort können:

- Zielordner,
- Dateiname

frei gewählt werden.

Firefox und Safari

Unterstützt der Browser diese Funktion nicht, wird die erzeugte Word-Datei automatisch über den normalen Download-Mechanismus des Browsers gespeichert.

Die Datei landet dann im entsprechenden Download-Ordner.

Auch in diesem Fall findet keine zusätzliche Übertragung der Datei statt.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Convert läuft vollständig lokal im Browser.

Es gibt:

- keine Cloud-Anbindung,
- kein Benutzerkonto,
- keine Cookies,
- keine Übertragung der PDF-Datei an einen Server.

Die Verarbeitung findet ausschließlich im Arbeitsspeicher des geöffneten Browsers statt.

Damit können auch:

- vertrauliche Dokumente,
- technische Berichte,
- aus einem beschädigten System gerettete PDF-Dateien,
- lokal erzeugte Engineering-Berichte

in ein bearbeitbares Word-Dokument umgewandelt werden, ohne die Ausgangsdatei an einen externen Konvertierungsdienst zu übertragen.

Dies ist besonders für den Einsatz innerhalb von JUMBO BLOCK Rescue relevant.

Keine dauerhafte Arbeitssitzung

Convert ist kein Dokumenteditor und besitzt keine eigene dauerhafte Arbeitssitzung.

Die ausgewählte PDF-Datei wird verarbeitet und anschließend als neue `.docx`-Datei ausgegeben.

Es gibt keinen dokumentierten Convert-Arbeitsstand, der separat gespeichert und später wieder geöffnet werden müsste.

Die Verarbeitung findet im Arbeitsspeicher des Browsers statt.

Sprache

Convert unterstützt Deutsch und Englisch.

Die Sprachumschaltung erfolgt über das globale Menü der JUMBO BLOCK Engineering Suite.

Dabei werden:

- Beschriftungen,
- Statusmeldungen

zwischen Deutsch und Englisch umgeschaltet.

Die gewählte Oberflächensprache hat **keinen Einfluss auf den Inhalt des erzeugten Word-Dokuments**.

Convert übersetzt den PDF-Inhalt nicht.

Grenzen und Hinweise

Keine pixelgenaue 1:1-Konvertierung

Convert verfolgt ausdrücklich nicht das Ziel, die PDF-Vorlage optisch exakt zu reproduzieren.

Priorität besitzt die **Bearbeitbarkeit des Ergebnisses in Word**.

Kleinere Layoutabweichungen sind deshalb Bestandteil des gewählten Konvertierungsansatzes.

Mehrspaltige Layouts

Mehrspaltige Seitenlayouts – beispielsweise ein Zeitungslayout mit mehreren nebeneinanderliegenden Textspalten – werden nicht gesondert erkannt.

Dadurch kann der Text im erzeugten Word-Dokument in einer unerwarteten Lesereihenfolge erscheinen.

Kurze Tabellen

Die automatische Tabellenerkennung setzt mindestens drei aufeinanderfolgende, spaltenartig ausgerichtete Zeilen voraus.

Tabellen mit lediglich:

- einer Zeile,
- zwei Zeilen

werden nicht als echte Word-Tabelle rekonstruiert.

Sie werden stattdessen als normaler Fließtext mit sichtbaren Abständen übernommen.

Kopf- und Fußzeilen

Kopf- beziehungsweise Fußzeilen werden nur automatisch erkannt, wenn eine Zeile:

- auf mindestens zwei Seiten identisch wiederkehrt

oder

- einem Seitenzahl-Muster entspricht.

Einmalige oder von Seite zu Seite wechselnde Randtexte – beispielsweise Kapitelüberschriften – bleiben bewusst Bestandteil des normalen Fließtexts.

Vektorgrafiken

Als Pfade gespeicherte Vektorgrafiken werden nicht konvertiert.

Unterstützt werden ausschließlich eingebettete Rastergrafiken.

Keine OCR

Gescannte PDF-Dateien ohne eingebettete Textebene können mit Convert nicht in bearbeitbaren Text umgewandelt werden.

Convert besitzt **keine OCR-Texterkennung**.

Die Konvertierung basiert auf der vorhandenen Textschicht der PDF-Datei.

Ein reines Bild-PDF liefert diese Grundlage nicht.

Technische Einordnung in JUMBO BLOCK Rescue

Convert ergänzt JUMBO BLOCK Rescue um einen **vollständig lokalen PDF-zu-Word-Konverter**.

Die funktionale Zuordnung lautet:

Reader → PDF-Dokumente lesen

PDF Merge → PDF-Seiten strukturell zusammenführen

PDF Printer → PDF-Dokumente für den Druck aufbereiten

Convert → PDF-Inhalte für eine anschließende Word-Bearbeitung rekonstruieren

Dabei verfolgt Convert einen bewusst anderen Ansatz als eine optische PDF-Kopie:

Nicht die pixelgenaue Reproduktion, sondern die anschließende Bearbeitbarkeit besitzt Priorität.

Durch:

- Textextraktion,
- Kopf-/Fußzeilenerkennung,
- Absatzrekonstruktion,
- Tabellenerkennung,
- Übernahme eingebetteter Rasterbilder,
- Zuordnung von PDF-Schriften zu Word-Schriften

wird versucht, aus der statischen PDF-Struktur wieder ein praktisch bearbeitbares Dokument aufzubauen.

Die vollständig lokale Verarbeitung ist insbesondere im Rescue-Einsatz von Bedeutung, weil vertrauliche oder wiederhergestellte Dokumente nicht für eine Konvertierung an einen externen Dienst übertragen werden müssen.

Weiterführende Dokumentation

JUMBO BLOCK Convert – Benutzerhandbuch · Version 0.3.2

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Convert“ auf **https://jumboblock.org/**.

D.19 – Archive

Bereich: Archive
Programm: JUMBO BLOCK Archive
Baseline 20261001: 1.1.1 ff.

Zweck

JUMBO BLOCK Archive ist das vollständig lokal arbeitende Programm der JUMBO BLOCK Engineering Suite zum **Packen und Entpacken von Dateien und Ordnern**.

Die Bedienung ist bewusst einfach gehalten. Dateien, Ordner oder Archive können in das Programm gezogen werden; Archive erkennt dabei selbst, ob:

- vorhandene Dateien beziehungsweise Ordner gepackt

oder

- ein vorhandenes Archiv entpackt

werden sollen.

Für die eigentliche Verarbeitung werden:

- keine Dateien hochgeladen,
- kein Benutzerkonto,
- nach dem Laden der Anwendung keine Internetverbindung

benötigt.

Archive ist für Desktop, Tablet und Smartphone ausgelegt.

Rolle innerhalb der Engineering Suite

Archive übernimmt innerhalb der Engineering Suite die **lokale Archivierungs-, Verpackungs- und Entpackfunktion**.

Es dient insbesondere dazu:

- mehrere Dateien zusammenzufassen,
- vollständige Ordnerstrukturen zu verpacken,
- Dateien komprimiert weiterzugeben,
- ZIP-Dateien mit AES-256 zu schützen,
- vorhandene Archive zu untersuchen und zu entpacken,
- bestimmte Archivformate in ZIP umzuwandeln.

Archive arbeitet nicht mit der gemeinsamen `project.dbm` der projektspezifischen Engineering-Anwendungen.

Die jeweils ausgewählten Dateien und Archive bleiben die eigentlichen Datenquellen.

Unterstützte Formate

Entpacken: ZIP (auch mit Passwort), 7z, RAR, TAR, TAR.GZ, TAR.XZ, TAR.BZ2, GZ
Packen: ZIP (auch mit Passwort), TAR.GZ, TAR

RAR- und 7z-Archive können ausschließlich **entpackt**, jedoch nicht erstellt werden.

Technische Grundlage der Archivformate

Die Verarbeitung von:

- ZIP,
- TAR,
- GZIP

ist eigener Programmcode der JUMBO BLOCK Engineering Suite.

Für:

- 7z,
- RAR,
- TAR.XZ,
- TAR.BZ2

verwendet Archive die frei lizenzierte Bibliothek:

libarchive

unter BSD-Lizenz.

Die entsprechende Bibliothek wird erst geladen, wenn ein solches Archiv geöffnet wird.

Fehlt die Datei:

`vendor/libarchive/libarchive-wasm.js`

funktionieren:

- ZIP,
- TAR,
- GZIP

weiterhin.

Für die von libarchive abhängigen Formate erscheint dann ein entsprechender Hinweis.

Dateien und Ordner packen

Zum Erstellen eines neuen Archivs stehen drei Wege zur Verfügung:

- **„Dateien packen“**
- **„Ordner packen“**
- Dateien beziehungsweise Ordner per Drag-and-drop in das Fenster ziehen.

Beim Packen eines Ordners bleibt die vorhandene Ordnerstruktur erhalten.

Auch **leere Ordner** werden übernommen.

Dateiliste

Vor dem eigentlichen Packen zeigt Archive die ausgewählten Dateien und Ordner einschließlich ihrer Größe an.

Über:

×

kann ein Eintrag wieder entfernt werden.

Mit:

„ + Dateien hinzufügen“

und:

„ + Ordner hinzufügen“

können weitere Inhalte ergänzt werden.

Zusätzliche Dateien und Ordner können außerdem jederzeit per Drag-and-drop hinzugefügt werden.

Archivname

Das Feld:

„Name des Archivs“

wird automatisch sinnvoll vorbelegt.

Beim Packen eines Ordners kann beispielsweise dessen Name als Ausgangspunkt verwendet werden.

Der vorgeschlagene Archivname kann vor dem Speichern geändert werden.

Daneben wird das gewünschte Archivformat gewählt.

Für den allgemeinen Austausch empfiehlt das Handbuch:

ZIP

Kompression

Beim Erstellen von ZIP-Dateien verwendet Archive das Kompressionsverfahren:

Deflate

Die Kompression erfolgt durch den Browser.

Bereits komprimierte Dateitypen werden nicht erneut komprimiert, wenn dadurch keine sinnvolle weitere Größenreduzierung zu erwarten ist.

Dies betrifft beispielsweise:

- Fotos,
- Videos,
- Musik,
- Archive,
- Office-Dokumente.

Unter:

„Erweitert“

kann die Kompression bei ZIP vollständig deaktiviert werden.

Das Packen erfolgt dadurch schneller, die resultierende Archivdatei wird jedoch größer.

Dateinamen und Zeitangaben

Dateinamen werden in:

UTF-8

gespeichert.

Damit bleiben beispielsweise deutsche Umlaute erhalten.

Beim Packen werden außerdem:

- Änderungsdatum,
- Uhrzeit

der Dateien übernommen.

ZIP64

Überschreitet ein ZIP-Archiv:

- 4 GB

oder enthält es mehr als:

- 65.535 Einträge,

verwendet Archive automatisch:

ZIP64

Damit können auch ZIP-Archive erzeugt werden, die die klassischen ZIP-Grenzen überschreiten.

Passwortschutz mit AES-256

Bei ZIP-Dateien kann:

„🔒 Mit Passwort schützen“

aktiviert werden.

Das Passwort muss zweimal eingegeben werden.

Über:

„Passwort anzeigen“

kann es zur Kontrolle sichtbar gemacht werden.

Die Verschlüsselung erfolgt mit:

AES-256 nach dem WinZip-Verfahren.

Das Handbuch empfiehlt ein Passwort mit mindestens:

12 Zeichen.

Kompatibilität verschlüsselter ZIP-Dateien

Die von Archive erzeugten AES-verschlüsselten ZIP-Dateien können unter anderem mit folgenden Programmen geöffnet werden:

- JUMBO BLOCK Archive,
- 7-Zip,
- WinZip,
- WinRAR,
- Keka,
- The Unarchiver.

Der in Windows integrierte ZIP-Ordner kann AES-verschlüsselte ZIP-Dateien dagegen nicht öffnen.

Sichtbare Metadaten bei ZIP-Verschlüsselung

Die AES-Verschlüsselung schützt den **Inhalt der Dateien**.

Wie bei ZIP üblich bleiben jedoch unter anderem:

- Dateinamen,
- Ordnerstruktur,
- Dateigrößen

ohne Passwort sichtbar.

Vertrauliche Informationen sollten deshalb nicht bereits in Dateinamen oder Ordnernamen enthalten sein.

Umgang mit Passwörtern

Archive speichert die verwendeten Passwörter nicht.

Ohne das korrekte Passwort ist der verschlüsselte Inhalt nicht wiederherstellbar.

Das Passwort sollte deshalb getrennt vom Archiv aufbewahrt werden.

Das Handbuch nennt als mögliches Beispiel:

JUMBO BLOCK Vault

Archiv und Passwort sollten außerdem nicht über denselben Übertragungsweg weitergegeben werden.

Archive öffnen und prüfen

Ein vorhandenes Archiv kann über:

„Archiv entpacken“

ausgewählt oder direkt in das Fenster gezogen werden.

Nach dem Laden zeigt Archive:

- Archivformat,
- Anzahl der Dateien,
- entpackte Gesamtgröße,
- vollständige Inhaltsliste.

Damit kann der Inhalt eines Archivs bereits vor dem eigentlichen Entpacken geprüft werden.

Alles entpacken

Über:

„Alles entpacken“

wird der vollständige Archivinhalt ausgegeben.

Chrome und Edge

In Chrome und Edge kann ein Zielordner gewählt werden.

Archive legt darin einen neuen Ordner mit dem Namen des Archivs an und stellt die enthaltene Ordnerstruktur wieder her.

Existiert bereits ein gleichnamiger Ordner, wird beispielsweise:

„Name (2)“

verwendet.

Vorhandene Dateien werden nicht überschrieben.

Firefox und Safari

In Browsern ohne entsprechenden Ordnerzugriff werden die Dateien einzeln in den Download-Ordner ausgegeben.

Unterordner werden dabei im Dateinamen durch:

„_“

angedeutet.

Bei einer größeren Anzahl von Dateien fragt Archive vorher nach.

Zusätzlich kann der Browser eine Freigabe für mehrere Downloads verlangen.

Zielordner und Browser-Schutz

Bei Browsern mit Ordnerzugriff startet die Ordnerauswahl im Download-Ordner.

Bestimmte sensible Bereiche werden vom Browser aus Sicherheitsgründen nicht als direktes Entpackziel zugelassen.

Das Handbuch nennt beispielsweise:

- den Download-Ordner selbst,
- Dokumente,
- Desktop,
- Benutzerverzeichnis,
- Systemlaufwerk wie `C:\`.

In einem solchen Fall kann innerhalb des gewünschten Bereichs ein Unterordner gewählt oder neu angelegt werden.

USB-Sticks und andere Laufwerke können direkt als Ziel gewählt werden.

Wird kein Zielordner gewählt, bietet Archive:

- **„Erneut Ordner wählen“**

oder

- **„Stattdessen herunterladen“**

an.

Einzelne Dateien entpacken

Neben jeder Datei innerhalb eines geöffneten Archivs steht:

„Speichern“

zur Verfügung.

Damit kann gezielt nur diese einzelne Datei entpackt werden.

Enthält ein Archiv lediglich eine Datei, wird diese bei:

„Alles entpacken“

direkt gespeichert.

Archive in ZIP umwandeln

Über:

„Als ZIP speichern“

können:

- 7z,
- RAR,
- TAR

in eine neue ZIP-Datei umgewandelt werden.

Damit kann aus einem Archivformat, das auf einem Zielsystem möglicherweise zusätzliche Software benötigt, ein weit verbreitetes ZIP-Archiv erzeugt werden.

Integritätsprüfung beim Entpacken

Archive prüft beim Entpacken jede Datei anhand der im Archiv vorhandenen Prüfinformationen.

Verwendet werden:

- CRC-32

beziehungsweise bei AES-geschützten ZIP-Dateien:

- ein Echtheitscode.

Wird eine Beschädigung des Archivs erkannt, zeigt Archive eine entsprechende Meldung an.

Diese Prüfung ist von einem externen kryptografischen Herkunftsnachweis zu unterscheiden. Sie dient der Erkennung von Beschädigungen beziehungsweise fehlerhaften Archivdaten.

Sichere Pfadbehandlung

Beim Entpacken bereinigt Archive die im Archiv enthaltenen Pfade.

Kein Archiveintrag darf außerhalb des gewählten Zielordners geschrieben werden.

Damit wird verhindert, dass manipulierte Pfadangaben den vorgesehenen Entpackbereich verlassen.

Zusätzlich werden:

- unter Windows unzulässige Zeichen durch _ ersetzt,

- symbolische Verknüpfungen,
- Sondereinträge

übersprungen.

Diese Schutzmaßnahmen sind insbesondere beim Öffnen fremder Archive innerhalb einer Rescue-Umgebung relevant.

Passwortgeschützte Archive öffnen

Erkennt Archive eine passwortgeschützte ZIP-Datei, erscheint ein entsprechender Hinweis.

Nach Eingabe des Passworts und Auswahl von:

„Weiter“

wird das Passwort unmittelbar geprüft.

Bei einem falschen Passwort erscheint eine Meldung.

Unterstützt werden verschlüsselte ZIP-Dateien mit:

- AES-128,
- AES-192,
- AES-256,

sowie das ältere ZIP-Verfahren:

ZipCrypto

Grenzen verschlüsselter 7z- und RAR-Archive

Verschlüsselte:

- 7z,
- RAR

können mit JUMBO BLOCK Archive nicht entpackt werden.

Das Programm weist in diesem Fall darauf hin.

Das Handbuch verweist hierfür auf geeignete externe Programme wie:

- 7-Zip,
- WinRAR.

Fortschritt und Abbrechen

Bei längeren Vorgängen zeigt Archive:

- Fortschritt in Prozent,
- aktuell bearbeitete Datei.

Über:

„Abbrechen“

oder:

Esc

kann der laufende Vorgang beendet werden.

Wichtig beim Entpacken:

Bereits in einen Zielordner geschriebene Dateien bleiben auch nach einem Abbruch dort erhalten.

Ein Abbruch stellt den Zielordner also nicht automatisch auf seinen vorherigen Zustand zurück.

Lokale Arbeitsweise und Bedeutung für JUMBO BLOCK Rescue

Archive verarbeitet sämtliche Dateien ausschließlich auf dem verwendeten Gerät.

Es werden weder:

- Dateiinhalte,
- Dateinamen,
- Passwörter

übertragen oder extern gespeichert.

Damit steht innerhalb von JUMBO BLOCK Rescue ein allgemeines lokales Werkzeug zur Verfügung, um beispielsweise:

- gerettete Dateien zusammenzufassen,
- Ordnerstrukturen zu sichern,
- Daten für die Weitergabe zu verpacken,
- sensible Daten verschlüsselt zu archivieren,
- fremde Archive zu untersuchen,
- Archivdaten auf einen USB-Datenträger zu entpacken,
- RAR-, 7z- oder TAR-Inhalte in ein allgemein nutzbares ZIP-Archiv zu überführen.

Für diese Arbeiten ist nach dem Laden der Anwendung keine Internetverbindung erforderlich.

Bedeutung für gerettete Daten

Archive kann verwendet werden, um bereits identifizierte und gesicherte Daten in eine transportierbare Archivdatei zu überführen.

Dabei gilt weiterhin das allgemeine Rescue-Prinzip:

Archivieren ersetzt weder Datenrettung noch Vertrauensprüfung.

Das Verpacken einer Datei in:

- ZIP,
- TAR.GZ,
- TAR

ändert nichts an der fachlichen oder sicherheitstechnischen Bewertung ihres Inhalts.

Gerettete oder aus einem kompromittierten System stammende Dateien sollten deshalb vor einer weiteren Verwendung entsprechend geprüft werden.

Arbeitsspeicher und Archivgröße

Die praktisch verarbeitbare Datenmenge hängt vom verfügbaren Arbeitsspeicher des Geräts ab.

ZIP, TAR und GZIP

Diese Formate werden weitgehend stückweise verarbeitet.

Abhängig vom verwendeten Gerät können dadurch auch Datenmengen von mehreren Gigabyte möglich sein.

7z, RAR, TAR.XZ und TAR.BZ2

Diese Archive werden vollständig in den Arbeitsspeicher geladen.

Das Handbuch empfiehlt deshalb für diese Formate Archivgrößen von ungefähr:

1 bis 2 GB.

Bei größeren Archiven kann der verfügbare Arbeitsspeicher zum begrenzenden Faktor werden.

Änderungsdaten beim Entpacken

Bei:

- 7z,
- RAR,
- TAR.XZ,
- TAR.BZ2

werden die ursprünglichen Änderungsdaten der Dateien nicht übernommen.

Die entpackten Dateien erhalten stattdessen das aktuelle Datum.

Dies ist insbesondere bei forensischen oder dokumentarischen Arbeitsabläufen zu beachten, bei denen Dateizeitstempel relevant sein können.

Nicht unterstützte Archivvarianten

Nicht unterstützt werden:

- einzelne `.xz`-Dateien ohne TAR-Archiv,
- einzelne `.bz2`-Dateien ohne TAR-Archiv,
- mehrteilige Archive.

Als Beispiele für mehrteilige Archive nennt das Handbuch:

- `.001`,
- `.z01`,
- `.part1.rar`.

Seltene ZIP-Kompressionsverfahren

Selten verwendete Kompressionsverfahren innerhalb von ZIP können über libarchive verarbeitet werden, sofern das betreffende Archiv nicht verschlüsselt ist.

Das Handbuch nennt beispielsweise:

- Deflate64,
- BZIP2,
- LZMA innerhalb von ZIP.

Mobile Nutzung

Archive ist auch für mobile Geräte ausgelegt.

Auf Smartphones werden die großen Schaltflächen untereinander angeordnet.

Das Packen vollständiger Ordner ist auf bestimmten Mobilgeräten – beispielsweise einem iPhone – technisch nicht möglich.

In diesem Fall wird die entsprechende Schaltfläche ausgeblendet.

Sprache

Archive unterstützt Deutsch und Englisch.

Die Sprachumschaltung erfolgt über das gemeinsame Menü der JUMBO BLOCK Engineering Suite.

Die gewählte Sprache verändert die Bedienoberfläche, nicht jedoch:

- Dateiinhalte,
- Archivnamen,
- Archivstruktur.

Grenzen und Sicherheitshinweise

Archive ist ein Werkzeug zum Packen und Entpacken von Dateien.

Es führt keine inhaltliche Sicherheitsbewertung der archivierten Dateien durch.

Ein erfolgreich entpacktes Archiv bedeutet deshalb nicht, dass dessen Inhalt:

- vertrauenswürdig,
- schadsoftwarefrei,
- fachlich korrekt

ist.

Insbesondere Archive aus unbekannter oder nicht vertrauenswürdiger Quelle sollten im Rescue-Kontext zunächst kontrolliert behandelt und bei Bedarf mit den dafür vorgesehenen Analyse- und Sicherheitswerkzeugen geprüft werden.

AES-256 schützt den Dateiinhalt eines entsprechend erzeugten ZIP-Archivs, nicht jedoch dessen sichtbare Metadaten wie Dateinamen und Ordnerstruktur.

Das verwendete Passwort wird nicht gespeichert und kann durch Archive nicht wiederhergestellt werden.

Ein Abbruch während des Entpackens entfernt bereits geschriebene Dateien nicht automatisch.

Bei wichtigen Daten sollte unabhängig vom Archivprogramm eine zusätzliche Datensicherung vorhanden sein.

Technische Einordnung in JUMBO BLOCK Rescue

Archive ergänzt JUMBO BLOCK Rescue um ein **vollständig lokales Werkzeug für Archivierung, Kompression, Verschlüsselung und Entpacken**.

Die Kernfunktionen umfassen:

Entpacken: ZIP (auch mit Passwort), 7z, RAR, TAR, TAR.GZ, TAR.XZ, TAR.BZ2, GZ
Packen: ZIP (auch mit Passwort), TAR.GZ, TAR

Besonders für Rescue-Arbeiten sind zusätzlich die Schutzmechanismen beim Entpacken relevant:

- Bereinigung von Pfaden,
- kein Schreiben außerhalb des gewählten Zielordners,
- Überspringen von Symlinks und Sondereinträgen,
- Prüfung der Archivdaten,
- kein automatisches Überschreiben vorhandener Dateien bei der Ordnerausgabe.

Archive dient damit sowohl der praktischen Verpackung geretteter Daten als auch dem kontrollierten lokalen Öffnen vorhandener Archivdateien.

Die Verarbeitung bleibt dabei auf dem Rescue-System beziehungsweise dem verwendeten Datenträger und benötigt keine externe Cloud-Infrastruktur.

Weiterführende Dokumentation

JUMBO BLOCK Archive – Benutzerhandbuch · Version 1.1.1

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Archive“ auf **https://jumboblock.org/**.

D.20 – Files

Bereich: Dateien
Programm: JUMBO BLOCK Files
Baseline 20261001: 1.0.0 ff.

Zweck

JUMBO BLOCK Files ist der lokale Dateimanager der JUMBO BLOCK Engineering Suite.

Mit ihm lassen sich:

- lokale Ordner,
- USB-Sticks,
- externe Laufwerke

durchsuchen und verwalten.

Files ermöglicht insbesondere:

- Dateien und Ordner anzeigen,
- Dateien in einer Vorschau öffnen,
- nach Dateinamen suchen,
- Text- und Code-Dateien nach Inhalten durchsuchen,
- Dateien und Ordner anlegen,
- umbenennen,
- kopieren,
- verschieben,
- löschen,
- Dateien herunterladen.

Files läuft vollständig lokal im Browser und funktioniert auch, wenn die Engineering Suite direkt von einem USB-Stick gestartet wird.

Es werden keine Dateien hochgeladen.

Die Anwendung enthält keine Programmbibliotheken von Drittanbietern.

Rolle innerhalb der Engineering Suite

Files übernimmt innerhalb der Engineering Suite die **allgemeine lokale Dateiverwaltung**.

Während die anderen Anwendungen bestimmte Dokument- oder Datenformate bearbeiten, stellt Files den grundlegenden Zugriff auf die zugrunde liegende Datei- und Ordnerstruktur bereit.

Die funktionale Abgrenzung lautet beispielsweise:

Files → Dateien und Ordner organisieren

Archive → Dateien und Ordner packen beziehungsweise Archive entpacken

Reader → unterstützte Dokumente lesen

Text → Text- und Codedateien bearbeiten

Writer → formatierte Dokumente erstellen und bearbeiten

Files arbeitet nicht mit der gemeinsamen `project.dbm` als eigenes Projektformat. Eine `project.dbm` ist für Files lediglich eine Datei innerhalb des Dateisystems.

Eine direkte Übergabe an andere Programme der Engineering Suite über eine Funktion **„Öffnen mit“** ist in Version 1.0.0 nicht enthalten.

Browser und Zugriffsrechte

Webbrowser dürfen aus Sicherheitsgründen nicht selbstständig auf beliebige Bereiche eines Datenträgers zugreifen.

Ein Ordner muss deshalb zunächst über:

„ Ordner öffnen“

ausgewählt und für Files freigegeben werden.

Anschließend kann Files innerhalb dieses Ordners und seiner Unterordner arbeiten.

Chrome, Edge und Opera am Computer

Diese Browser bieten den vollständigen Funktionsumfang.

Beim Öffnen eines Ordners fragt der Browser, ob Änderungen erlaubt werden sollen.

Wird diese Berechtigung nicht erteilt, kann der betreffende Ordner nur gelesen werden.

Firefox, Safari und die meisten Smartphones

Hier steht laut Handbuch nur der Lesezugriff zur Verfügung.

Ordner können:

- durchsucht,
- Dateien in der Vorschau betrachtet,
- Dateien heruntergeladen

werden.

Änderungen am Dateisystem sind in diesem Modus nicht möglich.

Files weist beim Start darauf hin.

Lesen und Schreiben / Nur lesen

Oben rechts zeigt Files den aktuellen Zugriffsmodus an:

✎ **Lesen und Schreiben**

oder:

👁 **Nur lesen**

Funktionen, die im jeweiligen Modus nicht zur Verfügung stehen, werden automatisch ausgegraut.

Damit ist unmittelbar erkennbar, ob Files innerhalb des aktuell geöffneten Ordners Änderungen durchführen darf.

Geschützte Bereiche

Der Browser schützt bestimmte Systembereiche vor einem direkten Zugriff.

Systemordner wie beispielsweise:

`C:\Windows`

`Programme`

`AppData`

werden vollständig gesperrt.

Auch:

- Dokumente,
- Downloads,
- Desktop,
- Benutzerverzeichnis,
- Systemlaufwerk wie `C:\`

können nicht direkt geöffnet werden.

Der Browser kann in diesem Fall beispielsweise melden, dass der gewählte Ordner Systemdateien enthält.

Unterordner innerhalb dieser Bereiche lassen sich dagegen öffnen.

Bei Bedarf kann im Auswahlfenster zunächst ein neuer Unterordner angelegt werden.

USB-Sticks und externe Laufwerke können direkt geöffnet werden.

Diese browserbedingten Zugriffsbeschränkungen sind für die Einordnung von Files als Rescue-Dateimanager wichtig: Files ersetzt keinen Dateimanager mit unmittelbaren Betriebssystem- oder Root-Rechten.

Ordner öffnen

Über:

„ Ordner öffnen“

wird ein Ordner ausgewählt.

Alternativ kann ein Ordner auf die linke Leiste gezogen werden.

Unter:

„Geöffnete Ordner“

können mehrere Ordner gleichzeitig eingebunden sein.

So können beispielsweise gleichzeitig:

- ein USB-Stick,
- ein Ordner auf einem internen Datenträger

bereitstehen.

Der Verzeichnisbaum lässt sich über:

▸

und:

▾

auf- und zuklappen.

Ein Klick auf einen Ordner öffnet ihn.

Über:

×

wird ein Ordner lediglich aus der Liste der geöffneten Ordner entfernt.

Der Ordner selbst wird dabei nicht gelöscht.

Zuletzt geöffnete Ordner

Files merkt sich bis zu:

acht zuletzt geöffnete Ordner.

Ein Klick auf einen solchen Eintrag öffnet den Ordner erneut.

Der Browser fragt dabei gegebenenfalls erneut nach der erforderlichen Berechtigung.

Ist beispielsweise ein zuvor verwendeter USB-Stick nicht mehr angeschlossen, zeigt Files einen entsprechenden Hinweis.

Die Liste **„Zuletzt geöffnet“** wird im Browser des jeweiligen Computers gespeichert.

Sie wird **nicht auf dem USB-Stick** gespeichert.

Navigation

Die Pfadleiste oberhalb der Dateiliste zeigt den aktuellen Speicherort.

Jeder Bestandteil des Pfades kann angeklickt werden.

Mit:

⬆

beziehungsweise der:

Rücktaste

wird in den übergeordneten Ordner gewechselt.

Mit:

↻

beziehungsweise:

F5

wird die aktuelle Anzeige aktualisiert.

Dateiliste

Die Listenansicht zeigt:

- Name,
- Größe,
- Typ,
- Änderungsdatum.

Ein Klick auf eine Spaltenüberschrift sortiert nach dem entsprechenden Merkmal.

Ein weiterer Klick kehrt die Sortierreihenfolge um.

Ordner stehen unabhängig von der Sortierung immer oben.

Listen- und Kachelansicht

Über:

▦

wird zur Kachelansicht gewechselt.

Für Fotos werden dort Miniaturbilder angezeigt.

Über:

☰

wird wieder zur Listenansicht zurückgekehrt.

Dateien und Ordner auswählen

Ein einfacher Klick wählt ein Element aus.

Für Mehrfachauswahlen stehen die üblichen Verfahren zur Verfügung:

Strg + Klick
einzelne Elemente ergänzen oder aus der Auswahl entfernen

Umschalt + Klick
zusammenhängenden Bereich auswählen

Strg + A
alle Elemente auswählen

Auf Touch-Geräten wählt das erste Antippen ein Element aus.

Das zweite Antippen öffnet es.

Dateien und Ordner öffnen

Ein:

Doppelklick

oder:

Eingabetaste

öffnet einen Ordner beziehungsweise die Vorschau einer Datei.

Dateivorschau

Files kann verschiedene Dateitypen direkt im Browser anzeigen.

Unterstützt werden laut Handbuch:

- Bilder,
- PDF,
- Audio,
- Video,
- Textdateien,
- Codedateien.

Bei Textdateien wird das erste:

Megabyte

angezeigt.

Mit:

◀ / ▶

beziehungsweise den Pfeiltasten kann in der Vorschau zur vorherigen oder nächsten Datei gewechselt werden.

Über:

„Herunterladen“

kann eine Kopie der angezeigten Datei gespeichert werden.

Esc

schließt die Vorschau.

Statusleiste und Ordnergröße

Die Statusleiste zeigt:

- Anzahl der Einträge,
- Größe der Einträge,
- Anzahl und Größe der aktuellen Auswahl.

Ein Doppelklick auf diese Angabe berechnet die Gesamtgröße des:

- ausgewählten Ordners

beziehungsweise:

- aktuell geöffneten Ordners

einschließlich aller Unterordner.

Neuen Ordner anlegen

Über:

„ + Ordner“

wird ein neuer Ordner angelegt.

Files schlägt einen Namen vor, der anschließend geändert werden kann.

Neue Textdatei anlegen

Über:

„ + Textdatei“

wird eine neue leere Textdatei erzeugt.

Auch hier wird ein Dateiname vorgeschlagen und kann angepasst werden.

Umbenennen

Über:

„✏ Umbenennen“

beziehungsweise:

F2

wird das ausgewählte Element umbenannt.

Nicht zulässig sind die Zeichen:

\ / : * ? " < > |

Ebenfalls nicht zulässig sind:

- ein Punkt am Ende,
- ein Leerzeichen am Ende

des Namens.

Kopieren

Zum Kopieren werden die gewünschten Elemente ausgewählt und anschließend:

„⧉ Kopieren“

beziehungsweise:

Strg + C

verwendet.

Danach wird der Zielordner geöffnet.

Dieser kann sich auch in einem anderen geöffneten Ordner oder auf einem anderen Datenträger befinden.

Mit:

„♢ Einfügen“

beziehungsweise:

Strg + V

werden die Elemente in das Ziel kopiert.

Verschieben

Zum Verschieben dient:

„✂ Ausschneiden“

beziehungsweise:

Strg + X

Anschließend wird der Zielordner geöffnet und über:

„📋 Einfügen“

beziehungsweise:

Strg + V

eingefügt.

Ausgeschnittene Elemente werden bis zum Einfügen blass dargestellt.

Die Zwischenablage gilt ausschließlich **innerhalb von Files**.

Kopieren zwischen Datenträgern

Da mehrere Ordner gleichzeitig geöffnet werden können, lassen sich Dateien beispielsweise zwischen:

- internem Datenträger,
- USB-Stick,
- externem Laufwerk

kopieren beziehungsweise verschieben, sofern der Browser für beide Bereiche die notwendigen Zugriffsrechte besitzt.

Dies ist eine der zentralen Funktionen von Files innerhalb von JUMBO BLOCK Rescue.

Namenskonflikte

Wird eine Datei in denselben Ordner kopiert, erzeugt Files automatisch eine Kopie mit dem Zusatz:

„- Kopie“

Existiert im Ziel bereits ein Element mit demselben Namen, stehen folgende Möglichkeiten zur Verfügung:

- **Ersetzen**
- **Beide behalten**
- **Überspringen**
- **Abbrechen**

Bei:

„Beide behalten“

wird dem Namen eine Nummer angehängt.

Mit:

„Für alle weiteren übernehmen“

kann dieselbe Entscheidung auf die übrigen Konflikte des laufenden Vorgangs angewendet werden.

Beim Ersetzen eines Ordners werden dessen Inhalte zusammengeführt.

Schutz vor rekursivem Kopieren

Ein Ordner kann nicht:

- in sich selbst,
- in einen seiner eigenen Unterordner

kopiert werden.

Files erkennt diesen Fall und verhindert den Vorgang mit einem entsprechenden Hinweis.

Drag-and-drop von außen

Dateien und Ordner, die beispielsweise aus einem anderen Dateimanager auf die Dateiliste gezogen werden, kopiert Files in den aktuell geöffneten Ordner.

Das Handbuch nennt als Beispiel den:

Windows-Explorer.

Fortschritt und Abbrechen

Längere Dateioperationen zeigen einen Fortschrittsdialog.

Der laufende Vorgang kann über:

„Abbrechen“

beendet werden.

Bereits vollständig kopierte Elemente bleiben dabei erhalten.

Eine Datei, die sich zum Zeitpunkt des Abbruchs gerade im Kopiervorgang befindet, wird nicht als halb geschriebene Datei zurückgelassen.

Löschen

Über:

„🗑 Löschen“

beziehungsweise:

Entf

wird die aktuelle Auswahl gelöscht.

Vorher erscheint eine Sicherheitsabfrage.

Besonders wichtig:

Files besitzt keinen Papierkorb.

Gelöschte Dateien und Ordner werden **endgültig entfernt**.

Wird ein Ordner gelöscht, gilt dies für seinen gesamten Inhalt.

Aus diesem Grund ist in der Sicherheitsabfrage:

„Abbrechen“

als hervorgehobene Voreinstellung vorgesehen.

Suchen

Die Suchfunktion durchsucht:

den aktuellen Ordner und alle Unterordner.

Dabei stehen verschiedene Sucharten zur Verfügung.

Suche nach Namen

Ein Teil des Namens genügt.

Groß- und Kleinschreibung spielen keine Rolle.

Unterstützt werden außerdem Platzhalter:

*****
beliebig viele Zeichen

?
genau ein Zeichen

Beispiele:

```
*.pdf
```

```
Bericht_20??.docx
```

Suche nach Typ

Über:

„Alle Typen“

kann die Suche auf bestimmte Kategorien eingeschränkt werden.

Zur Verfügung stehen:

- Ordner,
- Bilder,
- Dokumente,
- Audio,
- Video,
- Archive,
- Code/Text.

Die Typfilterung kann auch ohne zusätzlichen Suchbegriff verwendet werden.

Suche in Textinhalten

Mit:

„auch im Inhalt von Textdateien“

werden zusätzlich Text- und Codedateien nach ihrem Inhalt durchsucht.

Durchsucht werden Dateien bis zu einer Größe von:

10 MB.

Das Handbuch nennt beispielsweise:

- TXT,
- CSV,
- HTML,
- JSON,
- XML,
- Markdown,
- Quellcode.

Nicht im Inhalt durchsucht werden:

- Office-Dokumente,
- PDF-Dateien.

Suchergebnisse

Die Trefferliste zeigt zusätzlich den Speicherort jeder gefundenen Datei.

Angezeigt werden maximal:

5.000 Treffer.

Ein Doppelklick öffnet einen Treffer.

Mit:

„Suche beenden“

wird zur normalen Dateiansicht zurückgekehrt.

Eine laufende Suche kann abgebrochen werden.

Dateien herunterladen

Über:

⬇

werden die ausgewählten Dateien als Kopie im Download-Ordner des Browsers gespeichert.

Ordner können nicht unmittelbar heruntergeladen werden.

Sollen vollständige Ordner als einzelne Datei ausgegeben werden, verweist das Handbuch auf:

JUMBO BLOCK Archive

Damit ergänzen sich Files und Archive unmittelbar:

Files → Dateien und Ordner verwalten

Archive → vollständige Ordnerstrukturen in Archivdateien packen

Start vom USB-Stick

Wird die JUMBO BLOCK Engineering Suite von einem USB-Stick gestartet, arbeitet Files weiterhin mit Ordnern:

- auf dem Computer,
- auf dem USB-Stick selbst.

Am Computer muss dafür lediglich ein geeigneter Browser wie:

- Chrome,
- Edge

vorhanden sein.

Eine Installation von Files ist nicht erforderlich.

Die Freigabe eines Ordners gilt für den jeweiligen Browser auf dem jeweiligen Computer.

Wird derselbe Rescue-Stick an einem anderen Computer verwendet, müssen die gewünschten Ordner dort einmal neu ausgewählt und freigegeben werden.

Lokale Arbeitsweise und Datenschutz

Alle Dateien werden ausschließlich auf dem verwendeten Gerät verarbeitet.

Es werden weder:

- Dateiinhalte,
- Dateinamen

übertragen.

Lokal im Browser gespeichert werden lediglich:

- Einstellungen für Ansicht und Sortierung,
- die Liste **„Zuletzt geöffnet“**.

Damit bleibt die eigentliche Dateiverarbeitung lokal.

Bedeutung für JUMBO BLOCK Rescue

Files stellt innerhalb von JUMBO BLOCK Rescue eine zusätzliche browserbasierte Oberfläche für allgemeine Dateioperationen bereit.

Besonders relevant sind:

- Zugriff auf USB-Sticks und externe Laufwerke,
- gleichzeitiges Öffnen mehrerer Ordner,
- Kopieren zwischen unterschiedlichen Speicherorten,
- Vorschau wichtiger Dateitypen,
- Suche nach Dateien,
- Inhaltsuche in Text- und Codedateien,
- Ermittlung von Ordnergrößen,
- lokale Verarbeitung ohne Dateiübertragung.

Damit können beispielsweise bereits gerettete Daten:

- kontrolliert gesichtet,
- gesucht,
- sortiert,
- auf einen anderen Datenträger kopiert,
- für eine weitere Verarbeitung vorbereitet

werden.

Files selbst ist dabei **kein Datenrettungsprogramm**.

Die eigentliche Wiederherstellung beschädigter oder gelöschter Daten bleibt Aufgabe der dafür vorgesehenen Rescue-Werkzeuge aus Block A.

Dateizeitstempel und Attribute

Beim Kopieren erhalten die erzeugten Kopien:

das aktuelle Datum.

Dateiattribute und Zugriffsrechte werden nicht übernommen.

Dies ist insbesondere dann zu beachten, wenn ursprüngliche:

- Zeitstempel,
- Dateiattribute,
- Zugriffsrechte

für eine Dokumentation, Analyse oder forensische Betrachtung relevant sind.

Für eine beweissichernde 1:1-Sicherung ist Files daher nicht vorgesehen.

Verschieben

Das Verhalten beim Verschieben hängt von Browser und Datenträger ab.

Innerhalb eines Laufwerks erfolgt das Verschieben direkt, wenn der Browser dies unterstützt.

Andernfalls wird:

1. das Element kopiert,
2. anschließend das Original gelöscht.

Diese Unterscheidung ist bei großen Datenmengen und bei Rescue-Arbeiten zu berücksichtigen.

Leistungsgrenzen

Die Arbeitsgeschwindigkeit hängt unter anderem ab von:

- Datenträger,
- Browser,
- Anzahl der Dateien.

Sehr große Ordner mit vielen Tausend Dateien können deshalb beim:

- Öffnen,
- Suchen,
- Kopieren

entsprechend mehr Zeit benötigen.

Gesperrte Dateien

Dateien, die gerade von einem anderen Programm verwendet werden, können vom Betriebssystem beziehungsweise Browser gesperrt sein.

Files meldet einen solchen Zustand entsprechend.

Sprache und Darstellung

Die Oberfläche unterstützt:

- Deutsch,
- Englisch.

Die Sprachwahl erfolgt über:

DE / EN

im gemeinsamen Menü der Engineering Suite.

Auf Smartphones kann über:

☰

die Ordnerleiste ein- beziehungsweise ausgeblendet werden.

Grenzen und Sicherheitshinweise

Files arbeitet innerhalb der Sicherheitsgrenzen des Browsers.

Es besitzt keine direkten System- oder Root-Rechte.

Bestimmte geschützte Bereiche können deshalb nicht geöffnet werden.

Besonders zu beachten ist die Löschfunktion:

Es gibt keinen Papierkorb. Gelöschte Dateien und Ordner sind endgültig entfernt.

Vor einer Löschoperation muss daher eindeutig geprüft werden, ob:

- der richtige Datenträger,
- der richtige Ordner,
- die richtigen Dateien

ausgewählt sind.

Das allgemeine Rescue-Prinzip bleibt maßgebend:

Originale und beschädigte Quellen möglichst unverändert erhalten.

Files sollte bei Datenrettungsarbeiten deshalb vorzugsweise für bereits gesicherte Daten beziehungsweise Arbeitskopien eingesetzt werden, wenn Schreiboperationen erforderlich sind.

Beim Kopieren werden ursprüngliche Dateizeitstempel, Attribute und Zugriffsrechte nicht vollständig erhalten.

Eine direkte Übergabe an andere Anwendungen der Engineering Suite über **„Öffnen mit“** ist in Version 1.0.0 nicht vorhanden.

Wichtige Daten sollten unabhängig vom Dateimanager zusätzlich gesichert werden.

Wichtige Tastenkombinationen

Rücktaste / ⬆
übergeordneter Ordner

F5 / ↻
Ansicht aktualisieren

Doppelklick / Eingabe
Ordner öffnen beziehungsweise Dateivorschau

Strg + Klick
Einzelauswahl ergänzen oder entfernen

Umschalt + Klick
Bereich auswählen

Strg + A
alles auswählen

F2
umbenennen

Strg + C
kopieren

Strg + X
ausschneiden

Strg + V
einfügen

Entf
endgültig löschen – nach Sicherheitsabfrage

Pfeil auf / ab
Auswahl innerhalb der Liste bewegen

Pfeil links / rechts
in der Vorschau zur vorherigen beziehungsweise nächsten Datei wechseln

Esc
Vorschau oder Dialog schließen beziehungsweise Auswahl aufheben

Technische Einordnung in JUMBO BLOCK Rescue

Files bildet innerhalb der Engineering Suite die **allgemeine lokale Dateiverwaltungsschicht**.

Die funktionale Zuordnung lautet:

Files → Dateien und Ordner anzeigen, suchen, kopieren, verschieben und verwalten

Archive → Dateien und Ordner packen beziehungsweise Archive entpacken

Reader → Dokumente betrachten

Text → Text- und Codedateien bearbeiten

Für den Rescue-Einsatz ist besonders die Kombination mit den Werkzeugen aus Block A relevant:

Rescue-Werkzeug → Daten identifizieren beziehungsweise wiederherstellen

Files → gerettete Daten sichten, suchen und organisieren

Archive → ausgewählte Daten bei Bedarf verpacken beziehungsweise verschlüsseln

Files erweitert JUMBO BLOCK Rescue damit um einen vollständig lokalen Dateimanager, der auch direkt aus der vom USB-Stick gestarteten Engineering Suite verwendet werden kann.

Dabei bleibt die entscheidende Sicherheitsgrenze eindeutig:

Files verwaltet vorhandene Dateien. Es ersetzt weder eine sektorbasierte Datensicherung noch ein Datenrettungs-, Integritäts- oder Forensikwerkzeug.

Weiterführende Dokumentation

JUMBO BLOCK Files – Benutzerhandbuch · Version 1.0.0

Ausführliche Bedienung und technische Details: siehe separates JUMBO BLOCK Handbuch „Files“ auf **https://jumboblock.org/.**

Die Reise geht weiter

Mit **JUMBO BLOCK Rescue – Baseline 20261001** endet dieses Handbuch.

Die Entwicklung dahinter endet hier nicht.

Was mit der Frage begann, wie wir Infrastruktur besser vor Wasser schützen können, hat uns längst zu einer viel größeren Frage geführt:

Wie erhalten wir Handlungsfähigkeit, wenn gewohnte Systeme nicht mehr funktionieren?

Darauf gibt es keine einzelne Antwort.

Manchmal braucht es einen Retentionsraum unter einer versiegelten Fläche.

Manchmal ein belastbares Engineering.

Manchmal eine lokale Dokumentation, auf die auch ohne Cloud zugegriffen werden kann.

Und manchmal braucht es einen USB-Stick, von dem ein unabhängiges System startet, während der eigentliche Rechner nicht mehr starten will.

JUMBO BLOCK Rescue ist deshalb kein Endpunkt.

Es ist ein weiterer Baustein unserer Vorstellung davon, was **resiliente Infrastruktur** sein kann.

Die Baseline 20261001 dokumentiert den heutigen Stand: vier Werkzeugblöcke, 70 Programme und Funktionen und eine vollständig lokale Engineering-Umgebung für den Fall, dass der normale technische Alltag plötzlich nicht mehr normal ist.

Was morgen notwendig sein wird, wissen wir nicht.

Aber genau darum geht es bei Resilienz.

Nicht jede Störung vorherzusagen.

Sondern Strukturen, Werkzeuge und Wissen bereitzuhalten, mit denen man auch auf das reagieren kann, was niemand vorhergesehen hat.

Deshalb werden wir weiterentwickeln.

Weiter testen.

Weiter hinterfragen.

Und wahrscheinlich werden wir auch weiterhin Dinge bauen, bei denen man uns zunächst fragt:

„Warum macht ein Unternehmen für Regenwasserrückhaltung eigentlich so etwas?“

Unsere Antwort darauf ist inzwischen ziemlich einfach:

Weil Infrastruktur zusammengehört.

Wasser, Bauwerke, Energie, Kommunikation, Engineering, Daten und IT existieren im Ernstfall nicht in getrennten Schubladen.

Sie müssen gemeinsam funktionieren.

Und wenn etwas davon ausfällt, muss jemand in der Lage sein, weiterzumachen.

In diesem Sinne ist die Reise tatsächlich noch lange nicht zu Ende.

Block by Block.

JUMBO BLOCK
Resilienz endet nicht am Bauwerk.

Herzliche Grüße
Ihr Team JUMBO BLOCK

Impressum:

Entwicklung

JUMBO BLOCK
ZANNI GROUP
Brauhof 12
44866 Bochum

Telefon: +49 (0) 2327 4178 191
Fax: +49 (0) 2327 4178 192
E-Mail: mail@jumboblock.de

Vertrieb

AQUATO® Umwelttechnologien GmbH

Ernstmeierstraße 24
32052 Herford

Telefon: +49 (0) 5221 10 21 90
Fax: +49 (0) 5221 10 21 920
E-Mail: info@aquato.de

Registergericht: Amtsgericht Bochum
Handelsregisternummer: HRA 7687
Umsatzsteuer-Identifikationsnummer gemäß § 27a Umsatzsteuergesetz:
DE340292357