1.4 Flüchtige Informationen und Beweise sichern

1.4 Flüchtige Informationen und Beweise sichern

Bei einer Störung ist der erste Impuls häufig:

- Anwendung neu starten
- Dienst neu starten
- Computer neu starten
- Kabel abziehen
- DNS-Cache leeren
- Prozess beenden
- Update installieren
- Konfiguration zurücksetzen

Diese Maßnahmen können die Funktion wiederherstellen. Gleichzeitig können sie jedoch genau die Informationen vernichten, die zur Ermittlung der Ursache benötigt werden.

Deshalb gilt:

> “ Erst den aktuellen Zustand sichern, danach kontrolliert verändern.

Ein Neustart beseitigt unter anderem laufende Prozesse, bestehende Netzwerkverbindungen, Speicherinhalte, temporäre Zustände und Teile von Caches. Die Störung kann anschließend verschwunden sein, ohne dass ihre Ursache bekannt ist.

Ziel dieser Seite

Nach diesem Arbeitsschritt sollten:

- Fehlermeldung und genauer Zeitpunkt dokumentiert sein,
- aktuelle Prozesse und Dienste erfasst sein,
- Netzwerkzustand und Verbindungen erfasst sein,
- relevante Protokolle exportiert sein,
- wichtige Systemzustände dokumentiert sein,
- die gesammelten Dateien eindeutig zugeordnet sein,
- sensible Diagnosedaten geschützt sein,
- normale Störungsdiagnose und Sicherheitsvorfall unterschieden sein,
- erst anschließend kontrollierte Änderungen vorgenommen werden.

---

1. Das Stop–Sichern–Ändern-Prinzip

Phase	Tätigkeit
Stop	Nicht sofort neu starten, zurücksetzen oder bereinigen
Sichern	Flüchtige Zustände, Fehlermeldungen und Protokolle erfassen
Bewerten	Dringlichkeit und mögliches Sicherheitsrisiko beurteilen
Ändern	Eine kontrollierte Maßnahme durchführen
Prüfen	Ursprünglichen Fehler erneut testen
Dokumentieren	Maßnahme, Zeitpunkt und Ergebnis festhalten

Dieses Vorgehen verhindert, dass eine erfolgreiche Sofortmaßnahme die spätere Ursachenanalyse unmöglich macht.

---

2. Was sind flüchtige Informationen?

Flüchtige Informationen verändern sich während des laufenden Betriebs oder gehen beim Neustart verloren.

Information	Warum flüchtig?
Arbeitsspeicher	Inhalt geht beim Ausschalten oder Neustart verloren
Laufende Prozesse	Werden beendet und beim Start möglicherweise anders aufgebaut
Prozess-IDs	Werden nach einem Neustart oder Prozessstart neu vergeben
Netzwerkverbindungen	TCP-Sitzungen und temporäre Verbindungen werden beendet
Offene Dateien	Zuordnung zwischen Prozess und Datei geht verloren
ARP-/Neighbor-Cache	Wird automatisch aktualisiert oder beim Neustart geleert
DNS-Cache	Einträge laufen ab oder werden geleert
Routingzustand	Dynamische Routen können sich verändern
Angemeldete Benutzer	Sitzungen werden beendet
Temporäre Dateien	Können automatisch entfernt werden
Zwischenspeicher	Ändern sich durch normalen Betrieb
Auslastungswerte	CPU-, RAM-, Datenträger- und Netzwerklast ändern sich fortlaufend
Live-Protokollmeldungen	Können durch Logrotation oder begrenzte Speichergröße verschwinden
Fehlermeldung auf dem Bildschirm	Verschwindet nach Schließen der Anwendung

Information	Warum flüchtig?
Zustand eines hängenden Prozesses	Geht beim Beenden oder Neustart verloren

---

3. Reihenfolge nach Flüchtigkeit

RFC 3227 empfiehlt, Beweise grundsätzlich von den flüchtigsten zu den weniger flüchtigen Informationen zu sichern.

Für die praktische Störungsanalyse kann folgende Reihenfolge verwendet werden:

Priorität	Zu sichernde Information	Beispiel
1	Sichtbare Fehlermeldung und Uhrzeit	Screenshot, Fehlercode, betroffene Funktion
2	Laufende Prozesse und Arbeitsspeicherzustand	Prozessliste, CPU- und RAM-Auslastung
3	Aktive Netzwerkverbindungen	TCP-/UDP-Verbindungen, Quell- und Zieladressen
4	Temporäre Netzwerkinformationen	ARP-, Neighbor- und DNS-Cache
5	Angemeldete Benutzer und Sitzungen	Lokale, RDP-, SSH- oder Konsolensitzungen
6	Dienst- und Anwendungszustand	Laufende, gestoppte oder fehlgeschlagene Dienste
7	Aktuelle System- und Kernelmeldungen	Ereignisanzeige, Journal, Unified Log
8	Temporäre Dateien und Laufzeitdaten	PID-Dateien, Sockets, temporäre Verzeichnisse
9	Permanente Protokolle und Konfigurationen	Logdateien, Registry, Konfigurationsdateien
10	Zentrale und externe Daten	SIEM, Monitoring, Firewall, Cloud-Audit
11	Backups und Archivdaten	ältere Konfigurationen, Images, Sicherungen

Die konkrete Reihenfolge muss an den Vorfall angepasst werden. Bei einer laufenden Verschlüsselung oder Datenübertragung kann die Eindämmung wichtiger sein als eine vollständige Sammlung.

---

4. Normale Störung oder möglicher Sicherheitsvorfall?

Vor der Sammlung muss grob eingeschätzt werden, ob es sich um einen normalen technischen Fehler oder einen möglichen Sicherheitsvorfall handelt.

Normale technische Störung	Möglicher Sicherheitsvorfall
Dienst reagiert nicht	Unbekannter oder verdächtiger Prozess
Festplatte ist voll	Viele Dateien werden unerwartet verschlüsselt
DNS-Eintrag ist falsch	Unbekannte externe Netzwerkverbindungen
Update verursacht Kompatibilitätsfehler	Sicherheitssoftware meldet Schadsoftware
Anwendung ist abgestürzt	Neue unbekannte Administratorkonten
Netzwerkkabel ist defekt	Protokolle wurden gelöscht oder deaktiviert
Berechtigung wurde falsch gesetzt	Massenhafte fehlgeschlagene Anmeldungen
Zertifikat ist abgelaufen	Daten werden unerwartet nach außen übertragen
Systemressourcen sind ausgelastet	Sicherheitsrichtlinien wurden unerlaubt verändert
Konfiguration ist fehlerhaft	Ransomware-Nachricht oder Erpressung

Bei einer normalen Störung

- relevante Zustände sichern,
- anschließend kontrollierte Tests durchführen,
- pro Test möglichst nur eine Variable verändern,
- Ergebnisse dokumentieren.

Bei einem möglichen Sicherheitsvorfall

- Incident-Response-Prozess aktivieren,
- zuständige Sicherheitsverantwortliche informieren,
- keine unkoordinierten Änderungen durchführen,
- keine verdächtigen Dateien öffnen,
- keine Protokolle löschen,
- keine eigenständige Schadsoftwarebereinigung starten,
- System nicht unüberlegt ausschalten,
- alle eigenen Aktionen und Uhrzeiten dokumentieren,
- Eindämmungsmaßnahmen nach dem vorgesehenen Notfallplan durchführen.

> Bei laufender Verschlüsselung, aktiver Ausbreitung oder möglichem Datenabfluss kann eine schnelle Netzwerkisolation erforderlich sein. Die Entscheidung sollte nach dem Incident-Response-Plan und durch die zuständige Stelle erfolgen. Die Beweissicherung darf eine notwendige Eindämmung nicht gefährlich verzögern.

---

5. Was vor der Sicherung vermieden werden sollte

Maßnahme	Möglicher Informationsverlust
Computer neu starten	Prozesse, Arbeitsspeicher und Verbindungen gehen verloren
Dienst neu starten	ursprünglicher Dienstzustand und Prozess-ID gehen verloren
Anwendung schließen	Fehlermeldung und Prozesszustand verschwinden
Prozess beenden	offene Dateien, Verbindungen und Speicherzustand gehen verloren
`ipconfig /flushdns` ausführen	Windows-DNS-Cache wird geleert
ARP- oder Neighbor-Cache leeren	Zuordnung zwischen IP- und MAC-Adressen geht verloren
Protokolle leeren	Ereignisse werden dauerhaft entfernt
Datenträgerbereinigung starten	temporäre Dateien und mögliche Belege verschwinden
Update installieren	Ausgangszustand wird verändert
Konfiguration zurücksetzen	fehlerverursachende Einstellung ist nicht mehr nachvollziehbar
Snapshot zurückspielen	aktueller Systemzustand wird überschrieben
Sicherheitssoftware vollständig scannen lassen	Dateien können verändert, verschoben oder gelöscht werden
Diagnosewerkzeuge unkontrolliert installieren	Systemzustand und Zeitstempel verändern sich
Kabel sofort abziehen	Netzwerkzustand und Verbindungen gehen verloren
Browser oder Anwendung neu laden	Sitzung, Cache und Fehlermeldung können verändert werden

Diese Maßnahmen sind nicht grundsätzlich falsch. Sie sollten nur möglichst erst nach der Sicherung des relevanten Ausgangszustands erfolgen.

---

6. Kennzeichnung der Befehle

Kennzeichnung	Bedeutung
`[RO]`	Liest Informationen aus und soll keine Konfiguration verändern
`[FILE]`	Erstellt oder überschreibt eine Ausgabedatei
`[PRIV]`	Benötigt möglicherweise Administrator- oder Root-Rechte
`[ACTIVE]`	Startet eine aktive Messung oder Aufzeichnung
`[SENSITIV]`	Ausgabe kann vertrauliche oder personenbezogene Daten enthalten

> Auch ein `[RO]`-Befehl erzeugt einen Prozess und kann Spuren in Protokollen, Shell-Historien oder Zugriffsdaten hinterlassen. Für eine normale Störungsdiagnose ist das meist vertretbar. Bei einer forensischen Untersuchung müssen die freigegebenen Werkzeuge und Verfahren der Organisation verwendet werden.

---

7. Fehlermeldung vollständig sichern

Folgende Informationen sollten erfasst werden:

- vollständiger Fehlertext
- Fehlernummer oder Fehlercode
- Uhrzeit einschließlich Zeitzone
- betroffene Anwendung oder Funktion
- Benutzeraktion unmittelbar vor dem Fehler
- Benutzerkonto oder technisches Konto
- Hostname und IP-Adresse
- verwendete URL, Serveradresse oder Freigabe
- sichtbarer Anwendungszustand
- Häufigkeit und Reproduzierbarkeit
- betroffene und nicht betroffene Systeme
- verwendeter Screenshot-Dateiname
- zugehörige Ticketnummer

Guter Screenshot

Ein guter Screenshot zeigt:

- die vollständige Fehlermeldung,
- die Titelleiste der Anwendung,
- den sichtbaren Kontext,
- gegebenenfalls die Adresszeile,
- Datum und Uhrzeit,
- keine unnötigen geheimen Daten.

Ein zu stark zugeschnittener Screenshot kann wichtige Informationen wie Anwendung, Zielsystem oder Zeitpunkt entfernen.

Bei einem möglichen Sicherheitsvorfall

Wenn jede Interaktion mit dem System vermieden werden soll, kann der Bildschirm mit einem zweiten Gerät fotografiert werden. Dabei sind Datenschutz- und Unternehmensrichtlinien zu beachten.

---

8. Aktuellen Zeitpunkt zuerst dokumentieren

Windows

`[RO] Get-Date -Format o`

`[RO] (Get-Date).ToUniversalTime().ToString("o")`

`[RO] Get-TimeZone`

`[RO] w32tm /query /status`

Linux

`[RO] date --iso-8601=seconds`

`[RO] date -u "+%Y-%m-%dT%H:%M:%SZ"`

`[RO] timedatectl status`

macOS

`[RO] date "+%Y-%m-%dT%H:%M:%S%z"`

`[RO] date -u "+%Y-%m-%dT%H:%M:%SZ"`

`[RO] systemsetup -gettimezone`

Die Zeitprüfung ist wichtig, damit lokale Systemereignisse mit Firewall-, Server-, Cloud- und Monitoring-Protokollen verglichen werden können.

---

9. Grundzustand des Systems erfassen

Windows

Information	Befehl
Hostname	`[RO] hostname`
Betriebssystem	`[RO] Get-ComputerInfo`
Windows-Version kompakt	`[RO] Get-CimInstance Win32_OperatingSystem | Select-Object Caption, Version, BuildNumber, LastBootUpTime`
Betriebsdauer	`[RO] (Get-Date) - (Get-CimInstance Win32_OperatingSystem).LastBootUpTime`
Aktueller Benutzer	`[RO] whoami`
Benutzer und Gruppen	`[RO][SENSITIV] whoami /all`
Angemeldete Sitzungen	`[RO] quser`
Umgebungsvariablen	`[RO][SENSITIV] Get-ChildItem Env:`

Information	Befehl
Laufwerke	`[RO] Get-Volume`
Datenträgerbelegung	`[RO] Get-PSDrive -PSProvider FileSystem`

> Umgebungsvariablen können Token, Zugangsdaten, interne Pfade oder andere vertrauliche Werte enthalten. Sie sollten nur erfasst werden, wenn sie für die Störung relevant sind.

Linux

Information	Befehl
Hostname und System	`[RO] hostnamectl`
Kernel und Architektur	`[RO] uname -a`
Distribution	`[RO] cat /etc/os-release`
Betriebsdauer	`[RO] uptime`
Letzter Systemstart	`[RO] uptime -s`
Angemeldete Benutzer	`[RO] who -a`
Benutzeraktivität	`[RO] w`
Aktueller Benutzer	`[RO] id`
Dateisystembelegung	`[RO] df -hT`
Blockgeräte	`[RO] lsblk -f`
Arbeitsspeicher	`[RO] free -h`

---

macOS

Information	Befehl
Hostname	`[RO] hostname`
macOS-Version	`[RO] sw_vers`
Kernel und Architektur	`[RO] uname -a`
Systemübersicht	`[RO] system_profiler SPSoftwareDataType SPHardwareDataType`
Betriebsdauer	`[RO] uptime`
Letzter Systemstart	`[RO] sysctl -n kern.boottime`
Angemeldete Benutzer	`[RO] who`

Information	Befehl
Aktueller Benutzer	`[RO] id`
Dateisystembelegung	`[RO] df -h`
Arbeitsspeicher	`[RO] vm_stat`

---

10. Laufende Prozesse sichern

Eine Prozessliste sollte möglichst vor dem Beenden oder Neustarten einer Anwendung erfasst werden.

Windows PowerShell

Prozessübersicht

`[RO] Get-Process | Sort-Object CPU -Descending`

Prozess, übergeordneter Prozess und Befehlszeile

`[RO][SENSITIV] Get-CimInstance Win32_Process | Select-Object ProcessId, ParentProcessId, Name, ExecutablePath, CommandLine`

Prozess anhand der Prozess-ID untersuchen

`[RO] Get-Process -Id <PID> | Format-List *`

Prozesse mit hoher Speichernutzung

`[RO] Get-Process | Sort-Object WorkingSet64 -Descending | Select-Object -First 20 Name, Id, CPU, WorkingSet64`

Linux

Vollständige Prozessliste

`[RO][SENSITIV] ps auxww`

Prozessbaum

`[RO][SENSITIV] ps -ef --forest`

Prozesse nach CPU-Auslastung

`[RO] ps aux --sort=-%cpu | head -n 20`

Prozesse nach Speichernutzung

`[RO] ps aux --sort=-%mem | head -n 20`

Aktuellen Systemzustand einmalig ausgeben

`[RO] top -b -n 1`

Offene Dateien eines Prozesses

`[RO][PRIV][SENSITIV] lsof -p <PID>`

---

macOS

Vollständige Prozessliste

`[RO][SENSITIV] ps auxww`

Prozesse nach CPU-Auslastung

`[RO] top -l 1 -o cpu`

Prozesse nach Speichernutzung

`[RO] top -l 1 -o mem`

Offene Dateien eines Prozesses

`[RO][PRIV][SENSITIV] lsof -p <PID>`

---

11. Aktive Netzwerkverbindungen sichern

Netzwerkverbindungen sollten vor dem Beenden eines Prozesses, dem Trennen des Netzwerks oder einem Neustart gesichert werden.

Windows

TCP-Verbindungen

`[RO][PRIV] Get-NetTCPConnection`

TCP-Verbindungen mit zugehörigem Prozess

`[RO][PRIV] Get-NetTCPConnection | Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, State, OwningProcess`

UDP-Endpunkte

`[RO][PRIV] Get-NetUDPEndpoint`

Klassische Übersicht

`[RO][PRIV] netstat -ano`

Die letzte Spalte enthält unter Windows die Prozess-ID.

Prozess zu einer PID ermitteln

`[RO] Get-Process -Id <PID>`

Linux

TCP-, UDP- und lauschende Sockets

`[RO][PRIV] ss -tulpn`

Alle TCP-Verbindungen

`[RO][PRIV] ss -tanp`

Alle UDP-Endpunkte

`[RO][PRIV] ss -uanp`

Prozesse mit Netzwerkverbindungen

`[RO][PRIV][SENSITIV] lsof -nP -i`

Ohne Root-Rechte werden möglicherweise nicht alle Prozessinformationen angezeigt.

---

macOS

Aktive TCP-Verbindungen

`[RO] netstat -anv -p tcp`

UDP-Endpunkte

`[RO] netstat -anv -p udp`

Lauschende TCP-Prozesse

`[RO][PRIV] lsof -nP -iTCP -sTCP:LISTEN`

Bestehende TCP-Verbindungen

`[RO][PRIV] lsof -nP -iTCP -sTCP:ESTABLISHED`

---

12. Netzwerkkonfiguration und Caches sichern

Windows

Information	Befehl
Vollständige IP-Konfiguration	`[RO] ipconfig /all`
PowerShell-Netzwerkübersicht	`[RO] Get-NetIPConfiguration`
Routingtabelle	`[RO] route print`
PowerShell-Routen	`[RO] Get-NetRoute`
ARP-/Neighbor-Cache	`[RO] arp -a`
PowerShell-Neighbor-Tabelle	`[RO] Get-NetNeighbor`
DNS-Cache	`[RO][SENSITIV] ipconfig /displaydns`
PowerShell-DNS-Cache	`[RO][SENSITIV] Get-DnsClientCache`
Proxykonfiguration	`[RO] netsh winhttp show proxy`

Linux

Information	Befehl
IP-Adressen und Interfaces	`[RO] ip -details address show`
Linkzustand	`[RO] ip -details link show`
Alle Routingtabellen	`[RO] ip route show table all`
IPv6-Routen	`[RO] ip -6 route show table all`
Neighbor-Tabelle	`[RO] ip neigh show`
Resolverzustand	`[RO] resolvectl status`
DNS-Konfiguration	`[RO] cat /etc/resolv.conf`
NetworkManager-Verbindungen	`[RO][SENSITIV] nmcli connection show`
NetworkManager-Geräte	`[RO] nmcli device status`

`resolvectl` ist nur vorhanden, wenn die entsprechende systemd-Komponente verwendet wird.

---

macOS

Information	Befehl
Interfaces und IP-Adressen	`[RO] ifconfig -a`
Routingtabelle	`[RO] netstat -rn`
Standardroute	`[RO] route -n get default`
ARP-Cache	`[RO] arp -an`

Information	Befehl
DNS-Konfiguration	[RO][SENSITIV] scutil --dns
Netzwerkdienste	[RO] networksetup -listallnetworkservices
Hardwareports	[RO] networksetup -listallhardwareports
Proxykonfiguration	[RO] scutil --proxy

---

13. Dienstzustand sichern

Windows

Alle Dienste anzeigen

[RO] Get-Service | Sort-Object Status, Name

Detaillierte Dienstinformationen

[RO][SENSITIV] Get-CimInstance Win32_Service | Select-Object Name, State, StartMode, StartName, ProcessId, PathName

Nur beendete automatisch startende Dienste

[RO] Get-CimInstance Win32_Service | Where-Object {$_.StartMode -eq "Auto" -and $_.State -ne "Running"} | Select-Object Name, State, StartMode

---

Linux

Fehlgeschlagene Dienste

[RO] systemctl --failed

Laufende Dienste

[RO] systemctl list-units --type=service --state=running

Alle installierten Diensteinheiten

[RO] systemctl list-unit-files --type=service

Bestimmten Dienst untersuchen

[RO] systemctl status <Dienstname> --no-pager

---

macOS

Geladene launchd-Dienste

`[RO] launchctl list`

Nach einem Dienst suchen

`[RO] launchctl list | grep -i "<Suchbegriff>"`

---

14. Ressourcen- und Datenträgerzustand sichern

Windows

Information	Befehl
Laufwerke und freier Speicher	`[RO] Get-Volume`
Dateisystemlaufwerke	`[RO] Get-PSDrive -PSProvider FileSystem`
Speicherauslastung pro Prozess	`[RO] Get-Process | Sort-Object WorkingSet64 -Descending`
Datenträgerstatus	`[RO] Get-PhysicalDisk`
Ereignisse zu Datenträgern	`[RO] Get-WinEvent -FilterHashtable @{LogName="System"; ProviderName="disk"} -MaxEvents 50`

Linux

Information	Befehl
Arbeitsspeicher	`[RO] free -h`
Dateisystembelegung	`[RO] df -hT`
Inode-Belegung	`[RO] df -ih`
Blockgeräte	`[RO] lsblk -f`
Mounts	`[RO] findmnt`
Kernelmeldungen	`[RO][PRIV] dmesg --ctime`
Laufende I/O-Statistik	`[RO][ACTIVE] iostat -xz 1 5`

`iostat` ist nicht auf jeder Installation vorhanden und gehört üblicherweise zum Paket `sysstat`.

---

macOS

Information	Befehl
Arbeitsspeicherstatistik	`[RO] vm_stat`
Dateisystembelegung	`[RO] df -h`
Laufwerke und Partitionen	`[RO] diskutil list`
Datenträgerinformationen	`[RO] diskutil info <Datenträger>`

Information	Befehl
Laufende I/O-Statistik	[RO][ACTIVE] iostat -w 1 -c 5

---

15. Protokolle exportieren

Protokolle sollten möglichst in ihrem nativen Format gesichert werden. Dadurch bleiben zusätzliche Felder, Ereigniskennungen und Metadaten erhalten.

Windows-Ereignisprotokolle exportieren

Zuerst muss ein neuer und freigegebener Zielordner angelegt werden.

[FILE] $target = Join-Path "<Zielpfad>" "$env:COMPUTERNAME-$(Get-Date -Format 'yyyyMMdd-HHmmss')"

[FILE] New-Item -ItemType Directory -Path $target

Systemprotokoll exportieren

[FILE][PRIV] wevtutil epl System "$target\System.evtx"

Anwendungsprotokoll exportieren

[FILE][PRIV] wevtutil epl Application "$target\Application.evtx"

Sicherheitsprotokoll exportieren

[FILE][PRIV][SENSITIV] wevtutil epl Security "$target\Security.evtx"

PowerShell-Protokoll exportieren

[FILE][PRIV][SENSITIV] wevtutil epl "Microsoft-Windows-PowerShell/Operational" "$target\PowerShell-Operational.evtx"

Ereignisse nur aus einem bestimmten Zeitfenster als CSV sichern

[FILE] Get-WinEvent -FilterHashtable @{LogName="System"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"} | Select-Object TimeCreated, Id, ProviderName, LevelDisplayName, Message | Export-Csv "$target\System-Zeitfenster.csv" -NoTypeInformation -Encoding UTF8

> Das Sicherheitsprotokoll kann personenbezogene und sicherheitsrelevante Informationen enthalten und benötigt normalerweise erhöhte Berechtigungen.

Linux-Journal exportieren

Aktuellen Systemstart sichern

`[FILE] journalctl -b --no-pager --output=short-iso-precise > "<Zielpfad>/journal-current-boot.log"`

Vorherigen Systemstart sichern

`[FILE] journalctl -b -1 --no-pager --output=short-iso-precise > "<Zielpfad>/journal-previous-boot.log"`

Bestimmtes Zeitfenster sichern

`[FILE] journalctl --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00" --no-pager --output=short-iso-precise > "<Zielpfad>/journal-time-window.log"`

Kernelmeldungen sichern

`[FILE][PRIV] dmesg --ctime > "<Zielpfad>/dmesg.log"`

Dienstprotokoll sichern

`[FILE] journalctl -u <Dienstname> --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00" --no-pager --output=short-iso-precise > "<Zielpfad>/<Dienstname>.log"`

Die Umleitung mit `>` erstellt eine Datei oder überschreibt eine vorhandene Datei. Deshalb sollte immer ein neuer Zielordner verwendet werden.

---

macOS-Protokolle exportieren

Letzte Stunde als Text sichern

`[FILE][SENSITIV] log show --last 1h --style compact > "<Zielpfad>/macOS-unified-log.txt"`

Bestimmtes Zeitfenster sichern

`[FILE][SENSITIV] log show --start "2026-07-30 14:00:00" --end "2026-07-30 15:00:00" --style compact > "<Zielpfad>/macOS-time-window.log"`

Nur Fehler und Faults sichern

`[FILE][SENSITIV] log show --last 1h --predicate 'messageType == error OR messageType == fault' --style compact > "<Zielpfad>/macOS-errors.log"`

Systemdiagnose über die Oberfläche erstellen

`Aktivitätsanzeige → Weitere Optionen → Systemdiagnose`

Apple beschreibt diese Funktion als Möglichkeit, einen Diagnosebericht mit Informationen und Protokollen über den Mac zu erstellen.

> Ein Systemdiagnosebericht kann umfangreiche und vertrauliche Informationen enthalten. Er darf nur an freigegebenen Speicherorten abgelegt und an berechtigte Empfänger weitergegeben werden.

---

16. Kompakte Windows-Diagnosesammlung

Das folgende Beispiel erstellt einen neuen Diagnoseordner und schreibt den aktuellen Zustand in einzelne Dateien.

`<Zielpfad>` muss vorher durch einen freigegebenen Speicherort ersetzt werden.

```
$target = Join-Path "<Zielpfad>" "$env:COMPUTERNAME-$(Get-Date -Format 'yyyyMMdd-HHmmss')"
New-Item -ItemType Directory -Path $target

Get-Date -Format o |
    Out-File "$target\collection-start.txt" -Encoding utf8

Get-CimInstance Win32_OperatingSystem |
    Select-Object Caption, Version, BuildNumber, LastBootUpTime |
    Format-List |
    Out-File "$target\operating-system.txt" -Encoding utf8

Get-CimInstance Win32_Process |
    Select-Object ProcessId, ParentProcessId, Name, ExecutablePath, CommandLine |
    Export-Csv "$target\processes.csv" -NoTypeInformation -Encoding utf8

Get-Service |
    Sort-Object Status, Name |
    Export-Csv "$target\services.csv" -NoTypeInformation -Encoding utf8

Get-NetTCPConnection |
    Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, State, OwningProcess |
    Export-Csv "$target\tcp-connections.csv" -NoTypeInformation -Encoding utf8

Get-NetUDPEndpoint |
    Export-Csv "$target\udp-endpoints.csv" -NoTypeInformation -Encoding utf8

ipconfig /all |
    Out-File "$target\ipconfig-all.txt" -Encoding utf8
```

```
route print |
    Out-File "$target\routes.txt" -Encoding utf8

arp -a |
    Out-File "$target\arp-cache.txt" -Encoding utf8

ipconfig /displaydns |
    Out-File "$target\dns-cache.txt" -Encoding utf8

Get-Volume |
    Format-List |
    Out-File "$target\volumes.txt" -Encoding utf8

wevtutil epl System "$target\System.evtx"
wevtutil epl Application "$target\Application.evtx"

Get-Date -Format o |
    Out-File "$target\collection-end.txt" -Encoding utf8
```

Hinweise

- Die Sammlung schreibt Dateien, verändert aber keine beabsichtigte Systemkonfiguration.
- Einige Informationen benötigen eine administrative PowerShell.
- Nicht verfügbare Cmdlets können zu Fehlermeldungen führen.
- Die Ausgabe kann Benutzernamen, IP-Adressen, Prozessbefehle und interne Systeminformationen enthalten.
- Bei einem Sicherheitsvorfall darf dieses Beispiel nur verwendet werden, wenn es dem freigegebenen Incident-Response-Verfahren entspricht.

---

17. Kompakte Linux-Diagnosesammlung

`<Zielpfad>` muss durch einen freigegebenen Speicherort ersetzt werden.

```
case_dir="<Zielpfad>/$(hostname)-$(date +%Y%m%d-%H%M%S)"
mkdir -p -- "$case_dir"

date --iso-8601=seconds > "$case_dir/collection-start.txt"
uname -a > "$case_dir/kernel.txt"
cat /etc/os-release > "$case_dir/os-release.txt"
```

```
uptime > "$case_dir/uptime.txt"
who -a > "$case_dir/logged-in-users.txt"
ps auxww > "$case_dir/processes.txt"
ss -tulpn > "$case_dir/network-sockets.txt" 2>&1
ip -details address show > "$case_dir/ip-addresses.txt"
ip route show table all > "$case_dir/routes.txt"
ip neigh show > "$case_dir/neighbors.txt"
resolvectl status > "$case_dir/dns-status.txt" 2>&1
systemctl --failed --no-pager > "$case_dir/failed-services.txt"
free -h > "$case_dir/memory.txt"
df -hT > "$case_dir/filesystems.txt"
df -ih > "$case_dir/inodes.txt"
lsblk -f > "$case_dir/block-devices.txt"
journalctl -b --no-pager --output=short-iso-precise > "$case_dir/journal-current-boot.log"
dmesg --ctime > "$case_dir/dmesg.log" 2>&1
date --iso-8601=seconds > "$case_dir/collection-end.txt"
```

Hinweise

- Ohne Root-Rechte fehlen möglicherweise Prozessnamen bei Netzwerkverbindungen und Teile der Kernelmeldungen.
- Nicht jedes Linux-System verwendet systemd, `journalctl` oder `resolvectl`.
- Fehlerausgaben werden bei einigen Befehlen mit `2>&1` in dieselbe Datei geschrieben.
- Bestehende Dateien mit identischem Namen werden durch `>` überschrieben. Deshalb wird ein neuer Zeitstempelordner verwendet.

---

18. Kompakte macOS-Diagnosesammlung

`<Zielpfad>` muss durch einen freigegebenen Speicherort ersetzt werden.

```
case_dir="<Zielpfad>/$(hostname)-$(date +%Y%m%d-%H%M%S)"
mkdir -p -- "$case_dir"

date "+%Y-%m-%dT%H:%M:%S%z" > "$case_dir/collection-start.txt"
sw_vers > "$case_dir/macos-version.txt"
uname -a > "$case_dir/kernel.txt"
system_profiler SPSoftwareDataType SPHardwareDataType > "$case_dir/system-profile.txt"
uptime > "$case_dir/uptime.txt"
who > "$case_dir/logged-in-users.txt"
ps auxww > "$case_dir/processes.txt"
```

```
ifconfig -a > "$case_dir/interfaces.txt"
netstat -rn > "$case_dir/routes.txt"
netstat -anv > "$case_dir/network-connections.txt"
arp -an > "$case_dir/arp-cache.txt"
scutil --dns > "$case_dir/dns-configuration.txt"
scutil --proxy > "$case_dir/proxy-configuration.txt"
launchctl list > "$case_dir/launchd-services.txt"
vm_stat > "$case_dir/memory.txt"
df -h > "$case_dir/filesystems.txt"
log show --last 1h --style compact > "$case_dir/unified-log-last-hour.txt"
date "+%Y-%m-%dT%H:%M:%S%z" > "$case_dir/collection-end.txt"
```

Hinweise

- `system_profiler` und `log show` können je nach System einige Zeit benötigen.
- Die Unified-Log-Ausgabe kann personenbezogene und sicherheitsrelevante Daten enthalten.
- Für eine vollständige offizielle Systemdiagnose kann die Aktivitätsanzeige verwendet werden.
- Ohne erhöhte Berechtigungen können bestimmte Informationen fehlen.

---

19. Paketaufzeichnung nur gezielt einsetzen

Eine Paketaufzeichnung kann zeigen:

- ob eine Verbindung tatsächlich aufgebaut wird,
- welche IP-Adresse verwendet wird,
- ob DNS-Anfragen beantwortet werden,
- ob TCP-Verbindungen zurückgesetzt werden,
- ob Pakete mehrfach übertragen werden,
- ob ein TLS-Handshake fehlschlägt,
- ob ein Server nicht antwortet.

Sie kann jedoch auch vertrauliche Inhalte, interne IP-Adressen, Hostnamen, Benutzerkennungen und Sitzungsinformationen enthalten.

Vor einer Aufzeichnung müssen deshalb geklärt sein:

- technische Notwendigkeit,
- zulässige Schnittstelle,
- erlaubter Zeitraum,
- Datenschutz,
- Speicherort,
- Zugriffsberechtigung,

- Aufbewahrungsdauer.

Windows mit pktmon

Aufzeichnung starten

`[FILE][ACTIVE][PRIV][SENSITIV] pktmon start --capture --pkt-size 0 --file-name "<Zielpfad>\capture.etl"`

Aufzeichnung beenden

`[ACTIVE][PRIV] pktmon stop`

ETL-Datei in PCAPNG umwandeln

`[FILE][SENSITIV] pktmon etl2pcap "<Zielpfad>\capture.etl" --out "<Zielpfad>\capture.pcapng"`

Linux mit tcpdump

`[FILE][ACTIVE][PRIV][SENSITIV] sudo tcpdump -i any -nn -s 0 -w "<Zielpfad>/capture.pcap"`

Beenden mit `Strg + C`.

---

macOS mit tcpdump

`[FILE][ACTIVE][PRIV][SENSITIV] sudo tcpdump -i <Interface> -nn -s 0 -w "<Zielpfad>/capture.pcap"`

Beenden mit `Strg + C`.

Das richtige Interface kann vorher mit folgendem Befehl bestimmt werden:

`[RO] networksetup -listallhardwareports`

Eine ausführliche Paketanalyse wird später im Netzwerk-Kapitel behandelt.

---

20. Prüfsummen erstellen

Eine kryptografische Prüfsumme hilft zu erkennen, ob eine gesicherte Datei nachträglich verändert wurde.

> Eine Prüfsumme beweist nicht automatisch, woher eine Datei stammt. Sie dokumentiert den Zustand der Datei zum Zeitpunkt der Prüfsummenbildung und ermöglicht spätere Integritätsprüfungen.

Die Prüfsummen sollten erst erstellt werden, nachdem die Sammlung abgeschlossen wurde und keine weiteren Dateien hinzugefügt werden.

Windows

Prüfsumme einer Datei

`[RO] Get-FileHash "<Dateipfad>" -Algorithm SHA256`

Prüfsummen eines Diagnoseordners als separate CSV-Datei

`[FILE] Get-ChildItem "$target" -File -Recurse | Get-FileHash -Algorithm SHA256 | Export-Csv "$target-SHA256.csv" -NoTypeInformation -Encoding UTF8`

Linux

Prüfsumme einer Datei

`[RO] sha256sum "<Dateipfad>"`

Prüfsummen aller Dateien eines Diagnoseordners

`[FILE] (cd "$case_dir" && find . -type f -print0 | sort -z | xargs -0 -r sha256sum) > "${case_dir}-SHA256SUMS.txt"`

---

macOS

Prüfsumme einer Datei

`[RO] shasum -a 256 "<Dateipfad>"`

Prüfsummen aller Dateien eines Diagnoseordners

`[FILE] find "$case_dir" -type f -exec shasum -a 256 {} \; > "${case_dir}-SHA256SUMS.txt"`

---

21. Integrität später überprüfen

Windows

`[RO] Get-FileHash "<Dateipfad>" -Algorithm SHA256`

Die neu berechnete Prüfsumme wird mit der zuvor dokumentierten Prüfsumme verglichen.

Linux

`[RO] sha256sum -c "<SHA256SUMS-Datei>"`

macOS

Wenn das Prüfsummenformat kompatibel vorliegt:

`[RO] shasum -a 256 -c "<SHA256SUMS-Datei>"`

---

22. Sicheren Speicherort auswählen

Diagnosedaten sollten nicht unkontrolliert auf dem betroffenen System verbleiben.

Geeignete Ziele

- freigegebener Incident- oder Diagnoseordner
- geschützter Netzwerkbereich
- verschlüsselter administrativer Datenträger
- vorgesehenes Ticketsystem
- freigegebene Forensikplattform
- zentraler Logserver
- SIEM
- schreibgeschütztes Archiv

Ungeeignete Ziele

- privater USB-Stick
- private Cloud
- privates E-Mail-Konto
- Messenger
- öffentlicher Dateiübertragungsdienst
- frei zugängliche Netzwerkfreigabe
- unverschlüsselter mobiler Datenträger
- dauerhaftes Ablegen auf dem Desktop
- direktes Hochladen vollständiger Protokolle in öffentliche Foren

---

23. Mögliche sensible Inhalte

Datenquelle	Mögliche sensible Informationen
Prozessliste	Benutzernamen, Pfade, Befehlszeilen, Token
Netzwerkverbindungen	interne IP-Adressen, externe Ziele, verwendete Dienste
DNS-Cache	besuchte oder verwendete Hostnamen
Ereignisprotokolle	Benutzerkonten, Systeme, Anmeldeereignisse
Paketaufzeichnung	Nutzdaten, Sitzungen, Hostnamen, Identifikatoren
Konfigurationsdateien	Kennwörter, API-Schlüssel, Zertifikate

Datenquelle	Mögliche sensible Informationen
Umgebungsvariablen	Token, Schlüssel, Zugangsdaten
Browserdaten	URLs, Sitzungsdaten, Benutzerverhalten
Systemdiagnose	umfangreiche Hardware-, Benutzer- und Prozessinformationen
Cloud-Auditdaten	Benutzeraktionen, IP-Adressen, Ressourcennamen

Vor der Weitergabe muss geprüft werden:

- Wer darf die Daten sehen?
- Welche Informationen sind tatsächlich erforderlich?
- Müssen Daten geschwärzt werden?
- Darf eine Datei verändert oder muss eine Kopie erstellt werden?
- Welche Aufbewahrungsfrist gilt?
- Wann und wie müssen die Daten gelöscht werden?

Bei Beweismitteln sollte eine unveränderte Originalkopie erhalten bleiben. Geschwärzte Arbeitskopien werden davon getrennt erstellt.

---

24. Beweismittelkette dokumentieren

Bei einem möglichen Sicherheitsvorfall oder einer rechtlich relevanten Untersuchung muss nachvollziehbar bleiben, wer wann mit welchen Daten gearbeitet hat.

Feld	Dokumentation
Fall- oder Ticketnummer	Eindeutige Referenz
System	Hostname, Gerätetyp und gegebenenfalls Seriennummer
Eigentümer oder Fachbereich	Verantwortliche Organisationseinheit
Sammler	Name oder Administratorkonto der erfassenden Person
Beginn der Sammlung	Datum, Uhrzeit und Zeitzone
Ende der Sammlung	Datum, Uhrzeit und Zeitzone
Quelle	System, Datenträger, Anwendung oder Protokoll
Verwendete Befehle	Exakte Befehle und Parameter
Verwendete Werkzeuge	Name und Version
Zielmedium	Speicherort oder Datenträgerkennung
Dateiname	Eindeutiger Name
Prüfsumme	SHA-256-Wert
Übergabe	Von wem, an wen und wann

Feld	Dokumentation
Zugriff	Wer öffnete oder kopierte die Daten?
Zweck	Diagnose, Incident Response oder forensische Untersuchung
Veränderungen	Jede bekannte Veränderung dokumentieren

Beispiel für einen eindeutigen Dateinamen

`INC-2026-0042-SRV-DC01-System-20260730-151500.evtx`

Der Dateiname enthält:

- Fallnummer
- Systemname
- Inhalt
- Datum
- Uhrzeit

---

25. Prioritäten bei laufendem Produktionsausfall

Bei einem kritischen Produktionsausfall kann nicht unbegrenzt auf eine vollständige Sammlung gewartet werden.

Eine sinnvolle Mindestaufnahme besteht aus:

1. Fehlermeldung und Zeitpunkt
2. betroffenem System und Benutzer
3. Prozess- und Dienstzustand
4. Netzwerkverbindungen
5. IP-Konfiguration und Routing
6. Ressourcen- und Datenträgerzustand
7. relevanten Ereignissen kurz vor dem Fehler
8. ausgeführter Maßnahme
9. Zustand nach der Maßnahme

Anschließend kann eine begründete Wiederherstellungsmaßnahme durchgeführt werden.

Priorität	Entscheidung
Menschen oder physische Sicherheit betroffen	Sicherheit hat Vorrang
Aktiver Sicherheitsangriff	Eindämmung nach Incident-Response-Plan
Kritischer Produktionsstillstand	Mindestaufnahme, danach freigegebene Wiederherstellung
Einzelner Arbeitsplatz betroffen	Ausführlichere Diagnose meist möglich

Priorität	Entscheidung
Fehler gut reproduzierbar	Zustand sichern und kontrollierte Tests durchführen
Fehler selten oder nicht reproduzierbar	Vor einer Änderung besonders sorgfältig sichern

---

26. Nach jeder Änderung erneut sichern

Nach einer Maßnahme sollte nicht nur notiert werden, ob „es wieder geht“.

Zu dokumentieren sind:

- genaue Maßnahme,
- ausführende Person,
- Start- und Endzeitpunkt,
- betroffene Systeme,
- Ausgangszustand,
- Zustand unmittelbar danach,
- ursprünglicher Test,
- Testergebnis,
- neue Fehlermeldungen,
- mögliche Nebenwirkungen,
- Beobachtungszeitraum.

Vorher-Nachher-Tabelle

Prüfpunkt	Vor der Maßnahme	Nach der Maßnahme
Fehler reproduzierbar		
Dienstzustand		
Prozess-ID		
CPU-Auslastung		
Speichernutzung		
Netzwerkverbindung		
DNS-Auflösung		
Ereignisprotokoll		
Benutzerfunktion		
Monitoringstatus		

---

27. Typische Fehler bei der Beweissicherung

Fehler	Folge	Besseres Vorgehen

Sofortiger Neustart	Flüchtige Zustände gehen verloren	Mindestzustand zuerst sichern
Screenshot ohne Uhrzeit	Ereignis ist schwer einzuordnen	Zeit und Zeitzone dokumentieren
Nur Fehlermeldung abschreiben	Kontext und Details fehlen	Vollständigen Screenshot und Fehlercode sichern
Protokoll als Text kopieren	Metadaten können fehlen	Wenn möglich natives Format exportieren
Diagnose auf betroffenem System speichern	Daten können beim Ausfall verloren gehen	Freigegebenes externes Ziel verwenden
Originaldatei bearbeiten	Integrität ist nicht mehr nachvollziehbar	Original schützen, Arbeitskopie verwenden
Keine Prüfsumme erstellen	Spätere Veränderung schwer erkennbar	SHA-256 dokumentieren
Mehrere Maßnahmen gleichzeitig	Wirkung ist nicht zuordenbar	Eine Variable pro Test ändern
Daten unverschlüsselt versenden	Datenschutz- und Sicherheitsrisiko	Freigegebenen Übertragungsweg verwenden
Vollständige Logs öffentlich hochladen	Interne Daten werden offengelegt	Daten prüfen, minimieren und schwärzen
Root- oder Admin-Rechte unnötig verwenden	Größerer Eingriff in das System	Niedrigste erforderliche Berechtigung verwenden
Sicherheitsvorfall wie normalen Fehler behandeln	Spuren werden vernichtet oder Angriff breitet sich aus	Incident-Response-Prozess aktivieren
Sammlung verzögert notwendige Eindämmung	Schaden kann zunehmen	Sicherheit und Eindämmung priorisieren
Nur lokale Protokolle sichern	Zentrale Hinweise fehlen	SIEM, Firewall, Cloud und Monitoring einbeziehen
Shell-Historie enthält sensible Befehle	Zugangsdaten können offengelegt werden	Geheimnisse niemals direkt in Befehle schreiben

28. Dokumentationsvorlage

Feld	Eintrag
Ticket- oder Fallnummer	
Datum und Uhrzeit	
Zeitzone	
Erfasser	
Betroffenes System	
Hostname	
IP-Adresse	

Feld	Eintrag
Betriebssystem	
Betroffener Benutzer	
Sichtbare Fehlermeldung	
Fehlercode	
Letzter funktionierender Zustand	
Erste bekannte Störung	
Screenshot gesichert	Ja / Nein
Prozesse gesichert	Ja / Nein
Verbindungen gesichert	Ja / Nein
Netzwerkzustand gesichert	Ja / Nein
Dienstzustand gesichert	Ja / Nein
Ressourcenstatus gesichert	Ja / Nein
Protokolle exportiert	Ja / Nein
Paketaufzeichnung erstellt	Ja / Nein / Nicht erforderlich
Diagnoseordner	
Prüfsumme erstellt	Ja / Nein
SHA-256-Datei	
Sensible Daten enthalten	Ja / Nein / Unbekannt
Sicherheitsvorfall vermutet	Ja / Nein
Incident Response informiert	Ja / Nein / Nicht erforderlich
Erste durchgeführte Änderung	
Zeitpunkt der Änderung	
Ergebnis der Änderung	
Weitere Maßnahmen	

Kurzcheckliste

- [ ] Nicht sofort neu gestartet
- [ ] Fehlermeldung vollständig gesichert
- [ ] Uhrzeit und Zeitzone dokumentiert
- [ ] Hostname und Betriebssystem erfasst
- [ ] Angemeldete Benutzer erfasst

- [ ] Laufende Prozesse erfasst
- [ ] Dienstzustand erfasst
- [ ] Aktive Netzwerkverbindungen erfasst
- [ ] IP-Konfiguration erfasst
- [ ] Routingtabelle erfasst
- [ ] ARP- oder Neighbor-Tabelle erfasst
- [ ] DNS-Zustand erfasst
- [ ] Ressourcen- und Datenträgerzustand erfasst
- [ ] Relevante Protokolle exportiert
- [ ] Zentrale Protokollquellen berücksichtigt
- [ ] Normalen Fehler und Sicherheitsvorfall unterschieden
- [ ] Sensible Inhalte der Sammlung berücksichtigt
- [ ] Freigegebenen Speicherort verwendet
- [ ] Prüfsummen erstellt
- [ ] Sammlung eindeutig dem Ticket zugeordnet
- [ ] Erst danach eine kontrollierte Änderung durchgeführt
- [ ] Zustand vor und nach der Änderung verglichen
- [ ] Alle Maßnahmen mit Uhrzeit dokumentiert

Ergebnis dieses Arbeitsschrittes

Am Ende dieses Schrittes liegt ein dokumentierter Ausgangszustand vor. Dadurch kann auch nach einem Neustart, Dienstneustart oder einer Konfigurationsänderung nachvollzogen werden, was vor der Maßnahme auf dem System geschah.

Die wichtigste Regel lautet:

> Eine schnelle Wiederherstellung behebt möglicherweise die Störung. Eine vorherige Zustandssicherung ermöglicht zusätzlich die Ursachenanalyse.

Bei einem möglichen Sicherheitsvorfall gelten die Incident-Response-Vorgaben der Organisation. Eine normale Diagnosesammlung ersetzt keine professionelle forensische Sicherung.

Nächste Seite:
1.5 Arbeitshypothesen bilden und Prüfungen priorisieren

Offizielle Standards und Herstellerdokumentation

- NIST – SP 800-61 Revision 3: Incident Response Recommendations and Considerations
- RFC Editor – RFC 3227: Guidelines for Evidence Collection and Archiving
- Microsoft Learn – wevtutil
- Microsoft Learn – Get-NetTCPConnection
- Microsoft Learn – Packet Monitor
- freedesktop.org – journalctl
- Red Hat – Configuring basic system settings
- Apple Support – Systemdiagnose in der Aktivitätsanzeige erstellen
- Apple Support – Protokollmeldungen in der Konsole anzeigen
- Apple Support – Diagnoseberichte in der Konsole untersuchen