1.3 Zeitpunkt und letzte Änderungen untersuchen

1.3 Zeitpunkt und letzte Änderungen untersuchen

Viele Störungen treten kurz nach einer Änderung auf. Deshalb gehört die Frage „Was hat sich verändert?“ zu den wichtigsten Schritten einer systematischen Fehleranalyse.

Dabei gilt jedoch:

> Eine zeitliche Übereinstimmung ist ein Hinweis, aber noch kein Beweis für einen ursächlichen Zusammenhang.

Ein Update kurz vor einer Störung kann die Ursache sein. Es kann aber ebenso Zufall sein, während beispielsweise gleichzeitig eine Firewall-Regel, ein VLAN oder ein Zertifikat geändert wurde.

Ziel dieser Seite

Nach diesem Schritt sollten folgende Fragen beantwortet sein:

- Wann funktionierte das System zuletzt nachweislich?
- Wann trat der Fehler erstmals nachweislich auf?
- Was wurde innerhalb dieses Zeitfensters verändert?
- Wer oder welcher Prozess führte die Änderung aus?
- Welche Systeme sind von der Änderung betroffen?
- Passt die Änderung technisch zum Fehlerbild?
- Kann der Zusammenhang durch einen Test bestätigt oder widerlegt werden?

---

1. Das Störungszeitfenster bestimmen

Zuerst wird der Zeitraum eingegrenzt, in dem die Ursache wahrscheinlich entstanden ist.

Zeitpunkt	Bedeutung	Beispiel
Letzter bekannter funktionierender Zustand	Funktion wurde erfolgreich verwendet oder geprüft	Benutzer meldete sich um 13:50 Uhr erfolgreich an
Erste bekannte Störung	Fehler wurde erstmals sicher beobachtet	Anmeldung schlug um 14:35 Uhr fehl
Meldezeitpunkt	Störung wurde an den Support gemeldet	Ticket wurde um 14:47 Uhr erstellt
Beginn der Untersuchung	Technische Analyse wurde gestartet	Administrator begann um 15:05 Uhr

Zeitpunkt	Bedeutung	Beispiel
Störungszeitfenster	Zeitraum zwischen „funktioniert“ und „funktioniert nicht“	13:50 bis 14:35 Uhr

Je kleiner das Störungszeitfenster ist, desto gezielter können Änderungen, Ereignisse und Protokolle durchsucht werden.

Wichtige Unterscheidung

- „Der Benutzer hat den Fehler um 14:35 Uhr bemerkt“ bedeutet nicht automatisch, dass der Fehler um 14:35 Uhr entstanden ist.
- Eine ausgefallene nächtliche Sicherung kann beispielsweise erst am nächsten Morgen auffallen.
- Ein Zertifikat kann um Mitternacht ablaufen, obwohl der betroffene Dienst erst Stunden später verwendet wird.
- Ein DHCP-, Kerberos- oder DNS-Problem kann bereits bestehen, bevor ein Benutzer eine sichtbare Störung meldet.

---

2. Uhrzeit und Zeitzone prüfen

Bevor Ereignisse verschiedener Systeme verglichen werden, müssen Uhrzeit und Zeitzone geprüft werden.

Abweichende Systemzeiten können eine korrekte Ereignisreihenfolge vortäuschen oder verbergen. Besonders relevant ist dies bei:

- Active Directory und Kerberos
- Zertifikaten
- VPN-Verbindungen
- Firewalls
- Netzwerkkomponenten
- Virtualisierungssystemen
- Cloud-Diensten
- verteilten Anwendungen
- SIEM- und Monitoring-Systemen

Aktuelle Uhrzeit anzeigen

Betriebssystem	Befehl
Windows PowerShell	`[RO] Get-Date -Format o`
Windows Eingabeaufforderung	`[RO] echo %date% %time%`
Linux	`[RO] date "+%Y-%m-%d %H:%M:%S %Z"`
macOS	`[RO] date "+%Y-%m-%d %H:%M:%S %Z"`

Aktuelle UTC-Zeit anzeigen

Betriebssystem	Befehl
Windows PowerShell	`[RO] (Get-Date).ToUniversalTime().ToString("o")`
Linux	`[RO] date -u "+%Y-%m-%dT%H:%M:%SZ"`
macOS	`[RO] date -u "+%Y-%m-%dT%H:%M:%SZ"`

Zeitzone und Zeitsynchronisation prüfen

Betriebssystem	Befehl
Windows PowerShell	`[RO] Get-TimeZone`
Windows	`[RO] w32tm /query /status`
Windows	`[RO] w32tm /query /source`
Linux mit systemd	`[RO] timedatectl status`
Linux mit chrony	`[RO] chronyc tracking`
Linux mit chrony	`[RO] chronyc sources -v`
macOS	`[RO] systemsetup -gettimezone`
macOS	`[RO] sntp -d time.apple.com`

> `[RO]` bedeutet „Read-only“. Der Befehl liest Informationen aus und soll keine Konfiguration verändern.

Bei der Dokumentation sollte immer angegeben werden, ob eine Uhrzeit als lokale Zeit oder als UTC-Zeit notiert wurde.

---

3. Letzten Systemstart bestimmen

Ein Neustart kann eine Störung ausgelöst, beseitigt oder sichtbar gemacht haben. Manche Änderungen werden außerdem erst nach einem Neustart wirksam.

Betriebssystem	Befehl
Windows PowerShell	`[RO] Get-CimInstance Win32_OperatingSystem | Select-Object LastBootUpTime`
Windows Eingabeaufforderung	`[RO] systeminfo`
Linux	`[RO] uptime -s`
Linux	`[RO] who -b`
Linux	`[RO] last reboot`

Betriebssystem	Befehl
macOS	`[RO] sysctl -n kern.boottime`
macOS	`[RO] last reboot`
Alle Unix-artigen Systeme	`[RO] uptime`

Windows-Betriebsdauer berechnen

`[RO] (Get-Date) - (Get-CimInstance Win32_OperatingSystem).LastBootUpTime`

Typische Schlussfolgerungen

Beobachtung	Mögliche Bedeutung
Störung begann direkt nach einem Neustart	Dienststart, Treiber, Update, Abhängigkeit oder Startreihenfolge prüfen
System wurde unerwartet neu gestartet	Stromversorgung, Absturz, Watchdog, Hypervisor oder automatischer Neustart prüfen
System wurde lange nicht neu gestartet	Ausstehende Updates, Ressourcenprobleme oder nicht aktivierte Konfigurationsänderungen prüfen
Fehler verschwindet nach Neustart	Flüchtiger Zustand, Ressourcenleck, blockierter Dienst oder fehlerhafter Cache möglich
Fehler erscheint erst nach Neustart	Änderung wurde möglicherweise erst beim Systemstart aktiviert

---

4. Änderungen im Störungszeitfenster sammeln

Folgende Änderungsquellen sollten systematisch geprüft werden:

Bereich	Typische Änderungen
Betriebssystem	Updates, Neustarts, Treiber, Sicherheitsrichtlinien
Anwendungen	Installation, Update, Deinstallation, Konfigurationsänderung
Dienste	Start, Stopp, Absturz, Starttyp oder Dienstkonto geändert
Netzwerk	VLAN, Routing, Switchport, ACL, Firewall-Regel, NAT
DNS	Neuer, gelöschter oder geänderter Eintrag
DHCP	Scope, Reservierung, Option, Lease oder Relay geändert
Active Directory	Benutzer, Gruppe, Gruppenrichtlinie, Computerobjekt
Berechtigungen	NTFS-, Freigabe-, Rollen- oder Cloud-Berechtigung
Zertifikate	Ausstellung, Austausch, Ablauf oder Vertrauenskette
Storage	Datenträger, LUN, Mount, Berechtigung, Speicherplatz

Bereich	Typische Änderungen
Virtualisierung	Snapshot, Migration, Ressourcen oder virtuelle Netzwerke
Backup	Sicherung, Wiederherstellung, Agent- oder Repository-Änderung
Cloud	Sicherheitsgruppe, Rolle, Richtlinie, Netzwerk oder Ressource
Automatisierung	Skript, geplante Aufgabe, Cronjob, Pipeline
Hardware	Austausch, Firmware, Verkabelung oder Portwechsel
Externe Dienste	Provider-Störung, Wartung oder geänderte Schnittstelle

---

5. Ereignisse im relevanten Zeitfenster anzeigen

Für die folgenden Beispiele wird dieses Störungszeitfenster verwendet:

- Letzter bekannter funktionierender Zustand: 30.07.2026, 14:00 Uhr
- Erste bekannte Störung: 30.07.2026, 15:00 Uhr

Die Werte müssen an den tatsächlichen Vorfall angepasst werden.

Windows-Ereignisse mit PowerShell durchsuchen

Systemprotokoll innerhalb eines Zeitfensters

`[RO] Get-WinEvent -FilterHashtable @{LogName="System"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"}`

Anwendungsprotokoll innerhalb eines Zeitfensters

`[RO] Get-WinEvent -FilterHashtable @{LogName="Application"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"}`

Warnungen und Fehler anzeigen

`[RO] Get-WinEvent -FilterHashtable @{LogName="System"; Level=2,3; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"} | Select-Object TimeCreated, Id, ProviderName, LevelDisplayName, Message`

Level	Bedeutung
1	Kritisch
2	Fehler
3	Warnung
4	Information

Level	Bedeutung
5	Ausführlich

Ereignisse als übersichtliche Tabelle darstellen

`[RO] Get-WinEvent -FilterHashtable @{LogName="System"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"} | Sort-Object TimeCreated | Format-Table TimeCreated, Id, ProviderName, LevelDisplayName -AutoSize`

> Nicht jeder Fehler erzeugt ein Ereignis der Stufe „Fehler“. Auch Informationsereignisse können wichtige Hinweise auf Dienststarts, Updates oder Konfigurationsänderungen enthalten.

Linux-Ereignisse mit journalctl durchsuchen

Alle Ereignisse innerhalb eines Zeitfensters

`[RO] journalctl --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"`

Nur Warnungen und schwerwiegendere Meldungen

`[RO] journalctl --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00" -p warning`

Ereignisse eines bestimmten Dienstes

`[RO] journalctl -u <Dienstname> --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"`

Beispiel für SSH:

`[RO] journalctl -u sshd --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"`

Ereignisse des aktuellen Systemstarts

`[RO] journalctl -b`

Ereignisse des vorherigen Systemstarts

`[RO] journalctl -b -1`

Kernel-Meldungen des aktuellen Systemstarts

`[RO] journalctl -k -b`

---

macOS-Ereignisse durchsuchen

Systemprotokoll innerhalb eines Zeitfensters

`[RO] log show --start "2026-07-30 14:00:00" --end "2026-07-30 15:00:00" --style compact`

Nur Fehler und Störungen

`[RO] log show --start "2026-07-30 14:00:00" --end "2026-07-30 15:00:00" --predicate 'messageType == error OR messageType == fault' --style compact`

Ereignisse eines bestimmten Prozesses

`[RO] log show --start "2026-07-30 14:00:00" --end "2026-07-30 15:00:00" --predicate 'process == "<Prozessname>"' --style compact`

Beispiel für den Prozess `softwareupdated`:

`[RO] log show --last 2h --predicate 'process == "softwareupdated"' --style compact`

Alternativ können Protokolle über die Anwendung **Konsole** untersucht werden.

---

6. Betriebssystem- und Softwareupdates prüfen

Windows

Installierte Hotfixes anzeigen

`[RO] Get-HotFix | Sort-Object InstalledOn -Descending`

Ausgewählte Informationen darstellen

`[RO] Get-HotFix | Sort-Object InstalledOn -Descending | Select-Object InstalledOn, HotFixID, Description, InstalledBy`

Windows-Update-Ereignisse anzeigen

`[RO] Get-WinEvent -LogName "Microsoft-Windows-WindowsUpdateClient/Operational" -MaxEvents 100 | Select-Object TimeCreated, Id, LevelDisplayName, Message`

Windows-Update-Ereignisse innerhalb eines Zeitfensters

`[RO] Get-WinEvent -FilterHashtable @{LogName="Microsoft-Windows-WindowsUpdateClient/Operational"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"}`

Windows-Update-Protokolldatei erzeugen

`[RO] Get-WindowsUpdateLog`

Das Cmdlet erstellt aus den ETL-Dateien eine lesbare Windows-Update-Protokolldatei.

Grafischer Weg

`Einstellungen → Windows Update → Updateverlauf`

> `Get-HotFix` zeigt nicht zwingend jede installierte Aktualisierung an. Die Ausgabe sollte deshalb mit dem Windows-Updateverlauf, dem Ereignisprotokoll und gegebenenfalls dem eingesetzten Patchmanagement verglichen werden.

Debian und Ubuntu

APT-Transaktionsverlauf anzeigen

`[RO] less /var/log/apt/history.log`

DPKG-Aktivitäten anzeigen

`[RO] less /var/log/dpkg.log`

Installationen, Aktualisierungen und Deinstallationen suchen

`[RO] grep -E " install | upgrade | remove " /var/log/dpkg.log`

Ältere Protokolle können rotiert und komprimiert vorliegen:

`[RO] zgrep -hE " install | upgrade | remove " /var/log/dpkg.log*`

Red Hat Enterprise Linux, Rocky Linux, AlmaLinux und Fedora

DNF-Transaktionen anzeigen

`[RO] dnf history list`

Details einer Transaktion anzeigen

`[RO] dnf history info <Transaktions-ID>`

Letzte Transaktion untersuchen

`[RO] dnf history info last`

Installierte Pakete nach Installationszeit sortieren

`[RO] rpm -qa --last`

SUSE Linux Enterprise und openSUSE

Zypper-Verlauf anzeigen

`[RO] less /var/log/zypp/history`

Nur Paketinstallationen anzeigen

`[RO] grep "|install|" /var/log/zypp/history`

Nur Paketaktualisierungen anzeigen

`[RO] grep "|update|" /var/log/zypp/history`

Arch Linux

Paketmanager-Protokoll anzeigen

`[RO] less /var/log/pacman.log`

Paketänderungen suchen

`[RO] grep -E "\[ALPM\] (installed|upgraded|removed)" /var/log/pacman.log`

---

macOS

Verlauf der Softwareupdates anzeigen

`[RO] softwareupdate --history`

Installationsverlauf anzeigen

`[RO] system_profiler SPInstallHistoryDataType`

Grafischer Weg

`Systeminformationen → Software → Installationen`

Die Liste kann nach dem Installationsdatum sortiert werden.

---

7. Dienständerungen und Dienstabstürze prüfen

Windows

Besonders interessante Ereignisse des Service Control Managers:

Ereignis-ID	Typische Bedeutung
7031	Dienst wurde unerwartet beendet
7034	Dienst wurde unerwartet beendet

Ereignis-ID	Typische Bedeutung
7036	Dienst hat seinen Status geändert
7040	Starttyp eines Dienstes wurde geändert
7045	Ein neuer Dienst wurde installiert

Relevante Dienstereignisse abfragen

`[RO] Get-WinEvent -FilterHashtable @{LogName="System"; ProviderName="Service Control Manager"; Id=7031,7034,7036,7040,7045; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"} | Select-Object TimeCreated, Id, Message`

Aktuellen Zustand eines Dienstes anzeigen

`[RO] Get-Service -Name <Dienstname>`

Dienstkonfiguration anzeigen

`[RO] Get-CimInstance Win32_Service -Filter "Name='<Dienstname>'" | Select-Object Name, State, StartMode, StartName, PathName`

Linux

Status eines Dienstes anzeigen

`[RO] systemctl status <Dienstname>`

Protokoll eines Dienstes anzeigen

`[RO] journalctl -u <Dienstname> --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"`

Fehlgeschlagene Dienste anzeigen

`[RO] systemctl --failed`

Zeitpunkt der Dienstaktivierung anzeigen

`[RO] systemctl show <Dienstname> -p ActiveEnterTimestamp -p InactiveEnterTimestamp`

---

macOS

Geladene launchd-Dienste anzeigen

`[RO] launchctl list`

Nach einem bestimmten Dienst suchen

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

Prozessereignisse untersuchen

`[RO] log show --last 2h --predicate 'process == "<Prozessname>"' --style compact`

---

8. Geplante und automatisierte Änderungen prüfen

Nicht jede Änderung wird manuell durch einen Administrator ausgelöst. Häufige Ursachen sind:

- geplante Aufgaben
- Cronjobs
- Systemd-Timer
- Wartungsskripte
- Patchmanagement
- Softwareverteilung
- CI/CD-Pipelines
- Konfigurationsmanagement
- Gruppenrichtlinien
- Backup- und Cleanup-Jobs
- automatische Zertifikatserneuerung
- automatische Skalierung oder Cloud-Automatisierung

Windows

Geplante Aufgaben anzeigen

`[RO] Get-ScheduledTask`

Laufzeitinformationen anzeigen

`[RO] Get-ScheduledTask | Get-ScheduledTaskInfo | Sort-Object LastRunTime -Descending`

Aufgaben mit Fehlerergebnis suchen

`[RO] Get-ScheduledTask | Get-ScheduledTaskInfo | Where-Object {$_.LastTaskResult -ne 0}`

Task-Scheduler-Ereignisse anzeigen

`[RO] Get-WinEvent -LogName "Microsoft-Windows-TaskScheduler/Operational" -MaxEvents 100`

> Ein von `0` abweichender Rückgabewert ist ein Hinweis, muss aber anhand der jeweiligen Aufgabe interpretiert werden.

Linux

Systemd-Timer anzeigen

[RO] systemctl list-timers --all

Cronjobs des aktuellen Benutzers anzeigen

[RO] crontab -l

Systemweite Cron-Verzeichnisse prüfen

[RO] ls -la /etc/cron.d /etc/cron.hourly /etc/cron.daily /etc/cron.weekly /etc/cron.monthly

Cron-Ereignisse auf Debian und Ubuntu suchen

[RO] journalctl -u cron --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"

Cron-Ereignisse auf RHEL-kompatiblen Systemen suchen

[RO] journalctl -u crond --since "2026-07-30 14:00:00" --until "2026-07-30 15:00:00"

---

macOS

Geladene launchd-Aufträge anzeigen

[RO] launchctl list

Systemweite LaunchDaemons anzeigen

[RO] ls -la /Library/LaunchDaemons

Systemweite LaunchAgents anzeigen

[RO] ls -la /Library/LaunchAgents

Benutzerspezifische LaunchAgents anzeigen

[RO] ls -la ~/Library/LaunchAgents

---

9. Konfigurationsänderungen vergleichen

Wenn eine frühere Konfigurationsversion vorhanden ist, sollte sie mit dem aktuellen Zustand verglichen werden.

Windows PowerShell

[RO] Compare-Object (Get-Content "<Vorherige-Datei>") (Get-Content "<Aktuelle-Datei>")

Linux und macOS

`[RO] diff -u "<Vorherige-Datei>" "<Aktuelle-Datei>"`

Git-verwaltete Konfigurationen

`[RO] git status`

`[RO] git diff`

`[RO] git log --oneline --decorate -n 20`

`[RO] git log -p -- "<Datei>"`

Mögliche Quellen früherer Konfigurationen

- Konfigurationsmanagement
- Versionsverwaltung
- Systembackup
- VM-Snapshot
- Firewall-Konfigurationsbackup
- Switch-Konfigurationsarchiv
- Gruppenrichtlinien-Backup
- Dokumentation
- Export aus der Verwaltungsoberfläche
- gewünschter Zustand aus Infrastructure as Code

> Der aktuelle Zustand zeigt nur, wie das System jetzt konfiguriert ist. Ohne Auditierung, Versionsverwaltung oder Backup lässt sich daraus nicht zuverlässig ableiten, wann und durch wen eine Änderung vorgenommen wurde.

---

10. Änderungen an Netzwerkkomponenten prüfen

Bei Netzwerkstörungen sollten nicht nur Server und Clients betrachtet werden.

Komponente	Zu prüfende Änderungen
Switch	Portstatus, VLAN, Trunk, Port-Security, STP, Firmware
Router	Routing, Interface, NAT, ACL, dynamisches Routing
Firewall	Regelwerk, NAT, VPN, Objektgruppen, Zertifikate
WLAN-Controller	SSID, VLAN-Zuordnung, Authentifizierung, Funkkanal
DHCP-Server	Scope, Optionen, Reservierungen, Relay
DNS-Server	Einträge, Zonen, Weiterleitungen, Replikation
Load Balancer	Backend, Health Check, Zertifikat, Listener

Komponente	Zu prüfende Änderungen
Proxy	Ausnahmen, Authentifizierung, Zertifikat, Filterregel
NAC-System	Richtlinie, Geräteprofil, Quarantäne
Provider	Wartung, Routingänderung, Störung

Zu verwendende Informationsquellen

- Konfigurations- und Auditprotokoll des Gerätes
- AAA-, TACACS+- oder RADIUS-Protokoll
- Syslog-Server
- SIEM
- Netzwerkmanagementsystem
- Monitoring
- Konfigurationsbackup
- Change-Ticket
- Hersteller- oder Provider-Statusseite
- Wartungsankündigungen

Besonders wichtige Angaben

- Benutzer- oder Automationskonto
- Zeitpunkt
- Quell-IP-Adresse
- betroffene Komponente
- alter Wert
- neuer Wert
- Commit-, Revisions- oder Transaktions-ID
- zugehörige Ticketnummer

---

11. Active Directory und Gruppenrichtlinien prüfen

Mögliche relevante Änderungen:

- Benutzer wurde deaktiviert oder gesperrt
- Gruppenmitgliedschaft wurde geändert
- Computerobjekt wurde verschoben oder gelöscht
- Gruppenrichtlinie wurde geändert
- Gruppenrichtlinie wurde neu verknüpft
- Sicherheitsfilterung wurde verändert
- DNS- oder Replikationsproblem trat auf
- Dienstkonto oder Kennwort wurde geändert
- Berechtigung wurde entzogen
- Vertrauensstellung wurde verändert

Windows-Sicherheitsprotokoll nach Änderungen durchsuchen

`[RO] Get-WinEvent -FilterHashtable @{LogName="Security"; StartTime=[datetime]"2026-07-30 14:00:00"; EndTime=[datetime]"2026-07-30 15:00:00"} | Select-Object TimeCreated, Id, Message`

Der Zugriff auf das Sicherheitsprotokoll benötigt entsprechende Berechtigungen.

Resultierende Gruppenrichtlinien anzeigen

`[RO] gpresult /r`

Ausführlichen HTML-Bericht erstellen

`[RO] gpresult /h "<Zielpfad>\gpresult.html"`

> Das Erstellen der HTML-Datei ist keine reine Leseoperation, verändert aber keine Systemkonfiguration. Es wird lediglich eine Berichtsdatei geschrieben.

Zeitpunkt der letzten Richtlinienverarbeitung prüfen

`[RO] Get-WinEvent -LogName "Microsoft-Windows-GroupPolicy/Operational" -MaxEvents 100`

Für eine zuverlässige Nachverfolgung von Änderungen müssen die passenden Überwachungsrichtlinien bereits vor dem Vorfall aktiviert worden sein.

---

12. Cloud- und SaaS-Änderungen prüfen

Je nach Umgebung sind unter anderem folgende Protokolle relevant:

Plattform	Typische Quelle
Microsoft Entra ID	Überwachungsprotokolle und Anmeldeprotokolle
Microsoft 365	Einheitliches Überwachungsprotokoll
Microsoft Azure	Activity Log und Resource Logs
Amazon Web Services	AWS CloudTrail
Google Cloud	Cloud Audit Logs
VMware vCenter	Tasks und Events
Proxmox VE	Task History und Systemprotokoll
Backup-System	Job-, Audit- und Konfigurationsverlauf
Endpoint-Management	Geräte-, Richtlinien- und Bereitstellungsverlauf
Softwareverteilung	Deployment- und Installationsstatus

Zu prüfen sind insbesondere:

- Wer führte die Änderung aus?
- Wurde die Änderung manuell oder automatisiert ausgeführt?
- Von welcher Quell-IP kam die Aktion?
- Welche Ressource wurde verändert?
- Was war der vorherige Wert?
- Was ist der aktuelle Wert?
- War die Änderung erfolgreich?
- Gibt es eine zugehörige Ticket-, Job- oder Request-ID?

---

13. Zeitachse erstellen

Alle relevanten Ereignisse werden in einer gemeinsamen Zeitachse dokumentiert.

Zeit	Quelle oder System	Ereignis oder Änderung	Benutzer oder Prozess	Beleg	Bewertung
13:50	Client-PC	Anmeldung erfolgreich	Max Mustermann	Benutzerangabe	Letzter funktionierender Zustand
14:10	Windows Server	Sicherheitsupdate installiert	Patchmanagement	Updateverlauf	Zeitlich auffällig
14:31	Switch	Port in anderes VLAN verschoben	Admin-Konto	Auditprotokoll	Technisch passend
14:35	Client-PC	Anmeldung nicht mehr möglich	Max Mustermann	Ticket und Ereignisprotokoll	Erste bekannte Störung
14:42	Monitoring	Domänencontroller weiterhin erreichbar	Monitoring-System	Messwert	Serverausfall unwahrscheinlich
15:12	Testport	Anmeldung funktioniert	Testbenutzer	Kreuztest	Switchport als Fehlerbereich bestätigt

Dieses Beispiel zeigt:

- Das Windows-Update fand zwar vor der Störung statt.
- Die VLAN-Änderung liegt jedoch näher am ersten Fehler.
- Das Fehlerbild passt technisch zur VLAN-Änderung.
- Der Kreuztest über einen anderen Switchport bestätigt den Zusammenhang.
- Das Update war lediglich zeitlich auffällig, aber nicht die wahrscheinlichste Ursache.

---

14. Änderungen bewerten

Nicht jede gefundene Änderung besitzt dieselbe Bedeutung.

Kriterium	Frage
Zeitliche Nähe	Liegt die Änderung kurz vor dem ersten Fehler?
Technische Plausibilität	Kann die Änderung genau dieses Fehlerbild verursachen?
Betroffener Umfang	Entspricht der Änderungsumfang den betroffenen Benutzern oder Systemen?
Reproduzierbarkeit	Tritt der Fehler nach der Änderung zuverlässig auf?
Gegenprobe	Funktioniert es ohne die Änderung oder auf einem unveränderten Vergleichssystem?
Protokollbeleg	Gibt es passende Fehler, Warnungen oder Statusänderungen?
Abhängigkeiten	Betrifft die Änderung einen abhängigen Dienst oder ein vorgelagertes System?
Vergleichswerte	Funktionieren nicht geänderte Systeme weiterhin?
Bekanntes Problem	Ist das Verhalten vom Hersteller dokumentiert?

Praktische Priorisierung

Bewertung	Bedeutung
Hoch	Änderung liegt im Zeitfenster, passt technisch und wird durch Tests gestützt
Mittel	Änderung liegt im Zeitfenster und ist technisch möglich, aber noch nicht bestätigt
Niedrig	Änderung liegt nur zeitlich nahe, passt aber kaum zum Fehlerbild
Ausgeschlossen	Gegenprobe oder Messung widerlegt den Zusammenhang

---

15. Korrelation und Ursache unterscheiden

Nur Korrelation

- Update wurde am selben Tag installiert.
- Fehler wurde später gemeldet.
- Es gibt keine passende Fehlermeldung.
- Andere identisch aktualisierte Systeme funktionieren.
- Eine Deinstallation verändert das Fehlerbild nicht.

Wahrscheinlicher ursächlicher Zusammenhang

- Fehler trat unmittelbar nach der Änderung auf.
- Nur geänderte Systeme sind betroffen.
- Nicht geänderte Vergleichssysteme funktionieren.

- Protokolle zeigen passende Fehler.
- Der Fehler lässt sich reproduzieren.
- Eine kontrollierte Rücknahme beseitigt den Fehler.
- Erneutes Anwenden der Änderung erzeugt den Fehler wieder.

Die stärkste Bestätigung entsteht durch einen kontrollierten A/B-Test:

Zustand A	Zustand B	Schlussfolgerung
Änderung vorhanden, Fehler vorhanden	Änderung entfernt, Fehler verschwunden	Zusammenhang wahrscheinlich
Änderung vorhanden, Fehler vorhanden	Änderung entfernt, Fehler bleibt	Änderung vermutlich nicht ursächlich
Geändertes System gestört	Unverändertes Vergleichssystem funktioniert	Änderung oder Systemunterschied priorisieren
Beide Systeme gestört	Gemeinsame Abhängigkeit untersuchen	Lokale Änderung weniger wahrscheinlich

---

16. Vor einem Rollback beachten

Eine Änderung darf nicht allein aufgrund zeitlicher Nähe unüberlegt zurückgenommen werden.

Vor einem Rollback sollten folgende Punkte geklärt sein:

- Ist die Änderung dokumentiert?
- Ist der ursprüngliche Zustand bekannt?
- Existiert eine getestete Rückfallmöglichkeit?
- Werden Sicherheitslücken erneut geöffnet?
- Entstehen Abhängigkeiten zu anderen Änderungen?
- Ist eine Freigabe erforderlich?
- Sind Konfiguration, Protokolle und Beweise gesichert?
- Kann die Auswirkung des Rollbacks überwacht werden?
- Gibt es einen Wartungszeitraum?
- Kann das System nach dem Rollback vollständig getestet werden?

Sicheres Vorgehen

1. Aktuellen Zustand dokumentieren.
2. Relevante Protokolle und Konfigurationen sichern.
3. Rollback-Auswirkungen bewerten.
4. Freigabe einholen.
5. Möglichst nur eine Änderung zurücknehmen.
6. Den ursprünglichen Fehler erneut testen.
7. Abhängige Funktionen prüfen.
8. Ergebnis und Uhrzeit dokumentieren.
9. System anschließend weiter beobachten.

> Ein Rollback ist selbst eine Änderung und kann neue Störungen verursachen.

---

17. Häufige Fehler bei der Änderungsanalyse

Fehler	Folge	Besseres Vorgehen
Erstbeste Änderung wird beschuldigt	Falsche Ursache wird verfolgt	Technische Plausibilität und Gegenprobe prüfen
Nur der Server wird untersucht	Netzwerk-, Client- oder Cloud-Änderung bleibt unentdeckt	Gesamten Kommunikationsweg betrachten
Uhrzeiten werden ungeprüft verglichen	Falsche Ereignisreihenfolge	Uhrzeit, Zeitzone und Synchronisation prüfen
Nur Fehlerereignisse werden gelesen	Wichtige Informationsereignisse fehlen	Auch Starts, Stopps und Statusänderungen prüfen
Aktueller Zustand wird als Verlauf interpretiert	Zeitpunkt und Urheber bleiben unbekannt	Audit-, Versions- und Änderungsprotokolle verwenden
Mehrere Änderungen werden gleichzeitig zurückgenommen	Ursache kann nicht mehr zugeordnet werden	Möglichst eine Variable pro Test verändern
Protokolle werden erst nach Neustart gesichert	Flüchtige Informationen gehen verloren	Beweise vor Änderungen und Neustarts sichern
Benutzerangabe wird als exakter Fehlerbeginn behandelt	Suchzeitraum wird zu eng gewählt	Letzten sicheren Funktionstest und erste sichere Störung trennen
Nur geplante Änderungen werden geprüft	Automatische oder unautorisierte Änderung fehlt	Audit-, Deployment- und Automationsprotokolle einbeziehen
Provider oder SaaS wird vergessen	Externe Störung bleibt unberücksichtigt	Service Health und Anbieterstatus prüfen

---

18. Dokumentationsvorlage

Feld	Eintrag
Letzter bekannter funktionierender Zustand	
Quelle dieser Information	
Erste bekannte Störung	
Quelle dieser Information	
Verwendete Zeitzone	
Zeitabweichungen festgestellt	Ja / Nein
Letzter Systemstart	
Änderungen im Zeitfenster	

Feld	Eintrag
Verantwortlicher Benutzer oder Prozess	
Zugehöriges Change-Ticket	
Technische Plausibilität	Hoch / Mittel / Niedrig
Protokollbeleg vorhanden	Ja / Nein
Vergleichssystem geprüft	Ja / Nein
Gegenprobe durchgeführt	Ja / Nein
Rollback durchgeführt	Ja / Nein
Ergebnis	
Vermutete Ursache	
Ursache bestätigt	Ja / Nein
Noch offene Prüfungen	

Kurzcheckliste

- [ ] Letzten funktionierenden Zustand bestimmt
- [ ] Erste bekannte Störung bestimmt
- [ ] Meldezeitpunkt nicht mit Fehlerbeginn verwechselt
- [ ] Uhrzeit und Zeitzone aller beteiligten Systeme geprüft
- [ ] Letzte Neustarts geprüft
- [ ] Betriebssystemupdates geprüft
- [ ] Softwareinstallationen und Paketänderungen geprüft
- [ ] Dienststarts, Dienststopps und Abstürze geprüft
- [ ] Geplante Aufgaben und Automatisierungen geprüft
- [ ] Netzwerk- und Firewalländerungen geprüft
- [ ] DNS-, DHCP- und Active-Directory-Änderungen geprüft
- [ ] Cloud- und SaaS-Auditprotokolle geprüft
- [ ] Konfiguration mit früherem Zustand verglichen
- [ ] Gemeinsame Zeitachse erstellt
- [ ] Zeitliche Korrelation von bestätigter Ursache getrennt
- [ ] Ursache durch Messung, Vergleich oder Gegenprobe geprüft
- [ ] Beweise vor Neustart oder Rollback gesichert
- [ ] Ergebnisse im Ticket dokumentiert

Ergebnis dieses Arbeitsschrittes

Am Ende sollte eine nachvollziehbare Zeitachse vorliegen. Darin stehen der letzte funktionierende Zustand, der erste bekannte Fehler und alle technisch relevanten Änderungen dazwischen.

Eine Änderung gilt erst dann als wahrscheinliche Ursache, wenn sie:

- zeitlich zum Fehler passt,
- das Fehlerbild technisch erklären kann,
- den tatsächlich betroffenen Umfang erklärt und
- möglichst durch einen Vergleichs- oder Rücknahmetest bestätigt wurde.

Nächste Seite:
1.4 Flüchtige Informationen und Beweise sichern

Quellen und weiterführende Dokumentation

- Microsoft Learn – Get-WinEvent
- Microsoft Learn – Get-HotFix
- Microsoft Learn – Windows-Update-Protokolldateien
- Microsoft Learn – Problembehandlung bei Windows-Server-Updates
- Microsoft Learn – Windows-Ereignisanzeige
- freedesktop.org – journalctl
- Red Hat – Configuring basic system settings
- Apple Support – Anzeigen von Protokollmeldungen in der Konsole
- Apple Support – Installationsverlauf auf einem Mac anzeigen

---