4.13 Update-, Patch- und Paketverwaltungsfehler

Updates können in verschiedenen Phasen scheitern: bei der Erkennung, beim Herunterladen, bei der Signaturprüfung, während der Installation, beim Neustart oder erst nach der Aktivierung der neuen Komponenten.

Dabei muss zwischen Betriebssystemupdates, Sicherheitsupdates, Treibern, Firmware, Anwendungspaketen und verwalteten Unternehmensupdates unterschieden werden.

Typische Auswirkungen sind:

- Updates werden nicht gefunden oder angeboten.
- Der Download bleibt bei einem bestimmten Prozentwert stehen.
- Pakete können wegen Abhängigkeitsfehlern nicht installiert werden.
- Signaturen oder Zertifikate werden abgelehnt.
- Ein Neustart wird wiederholt angefordert.
- Ein Update wird nach dem Neustart zurückgerollt.
- Dienste starten nach dem Update nicht mehr.
- Paketdatenbanken oder Updatekomponenten sind beschädigt.
- Ein WSUS-, Repository-, Proxy- oder MDM-Server ist nicht erreichbar.
- Der Datenträger besitzt nicht genügend freien Speicherplatz.
- Ein anderes Installationsprogramm blockiert die Paketverwaltung.
- Das System verwendet eine nicht mehr unterstützte Version.
- Ein Update ist für die verwendete Architektur oder Version ungeeignet.

> **Wichtig:** Die Meldung „Update fehlgeschlagen“ bezeichnet nur das Ergebnis. Für die Diagnose müssen Updatequelle, Fehlerphase, Fehlercode, Systemzustand und Protokolle getrennt untersucht werden.

Kennzeichnung der Befehle

Kennzeichnung	Bedeutung
`[RO]`	Nur lesender Befehl
`[TEST]`	Führt eine aktive Prüfung aus
`[PRIV]`	Erhöhte Berechtigungen erforderlich
`[FILE]`	Liest Dateien oder Verzeichnisse
`[SENS]`	Ausgabe kann sensible Informationen enthalten
`[CHANGE]`	Verändert Cache, Konfiguration oder Systemzustand
`[DISRUPT]`	Kann Dienste unterbrechen oder einen Neustart erfordern

1. Updatefehler strukturiert aufnehmen

Vor einer Reparatur müssen die genaue Updateart und die Fehlerphase bestimmt werden.

Zu erfassende Informationen

- Welches Betriebssystem und welche Version werden verwendet?
- Welches konkrete Update oder Paket schlägt fehl?
- Welche Paket-, KB-, Build- oder Versionsnummer ist betroffen?
- Wird das Update direkt vom Hersteller oder intern bereitgestellt?
- Tritt der Fehler bei Erkennung, Download, Prüfung, Installation oder Neustart auf?
- Wird ein numerischer oder hexadezimaler Fehlercode angezeigt?
- Sind alle Systeme oder nur einzelne Geräte betroffen?
- Wurde das System kürzlich geklont, wiederhergestellt oder migriert?
- Ist ein Neustart ausstehend?
- Besteht ausreichend freier Speicherplatz?
- Stimmen Datum und Uhrzeit?
- Wird ein Proxy, VPN, WSUS, MDM oder internes Repository verwendet?
- Wurden Sicherheitssoftware oder Firewallregeln verändert?
- Ist die Betriebssystemversion noch unterstützt?
- Funktionierte die Aktualisierung zuvor?
- Welche Änderungen erfolgten unmittelbar vor dem Fehler?

Fehlerphasen

Phase	Typische Ursachen
Erkennung	Updatequelle, Richtlinie, MDM, Repository oder nicht unterstützte Version
Download	Netzwerk, DNS, Proxy, Firewall, CDN oder Speicherplatz
Signaturprüfung	Falsche Zeit, Zertifikat, Schlüsselbund oder beschädigter Download
Vorbereitung	Speicherplatz, Abhängigkeiten, Paketdatenbank oder Systemdateien
Installation	Gesperrte Dateien, laufende Dienste, Abhängigkeiten oder Berechtigungen
Neustart	Pending-Reboot-Zustand, Bootfehler, Treiber oder Firmware
Aktivierung	Dienststart, Konfigurationsmigration oder inkompatible Anwendung
Rollback	Installationsfehler, Bootproblem oder nicht erfüllte Voraussetzung

Dokumentationsregel

Der originale Fehlercode muss vollständig übernommen werden. Beschreibungen wie „irgendein Updatefehler“ reichen für eine gezielte Analyse nicht aus.

2. Vorbedingungen auf allen Betriebssystemen prüfen

Viele Updatefehler entstehen nicht in der Paketverwaltung selbst.

Prüfbereich	Warum er wichtig ist
Freier Speicherplatz	Updates benötigen Platz für Download, Entpacken, Installation und Rückfall
Datum und Uhrzeit	Signaturen, Zertifikate und TLS-Verbindungen sind zeitabhängig
DNS	Update- und Repositoryserver müssen aufgelöst werden
Internet beziehungsweise internes Netzwerk	Updatequelle muss erreichbar sein
Proxy	Systemdienste verwenden möglicherweise andere Proxyeinstellungen als Benutzer
Neustartstatus	Eine ältere Installation kann noch nicht abgeschlossen sein
Systemversion	Nicht mehr unterstützte Versionen erhalten möglicherweise keine Updates
Architektur	Paket muss zu x64, ARM64 oder einer anderen Architektur passen
Paketquelle	Repository, WSUS oder MDM muss gültige Metadaten bereitstellen
Signaturen	Paket und Metadaten müssen vertrauenswürdig sein
Berechtigungen	Installation benötigt normalerweise administrative Rechte
Sicherung	Fehlerhafte Updates können einen Rollback erforderlich machen

Schnellprüfung des Speicherplatzes

Windows:

```
[RO] Get-Volume |
    Select-Object DriveLetter, FileSystemLabel,
        HealthStatus, SizeRemaining, Size
```

Linux:

```
[RO] df -hT
```

macOS:

```
[RO] df -h
```

Zeit prüfen

Windows:

```
[RO] Get-Date -Format o
```

Linux:

```
[RO] timedatectl status
```

macOS:

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

3. Betriebssystemversion und Architektur ermitteln

Ein Update kann nur installiert werden, wenn Version, Build, Edition und Architektur unterstützt werden.

Windows

```
[RO] Get-ComputerInfo |
    Select-Object WindowsProductName, WindowsEditionId,
        WindowsVersion, OsBuildNumber, OsArchitecture
```

Kompakter:

```
[RO] Get-CimInstance Win32_OperatingSystem |
    Select-Object Caption, Version, BuildNumber, OSArchitecture
```

Linux

```
[RO][FILE] cat /etc/os-release
```

```
[RO] uname -r
```

```
[RO] uname -m
```

macOS

```
[RO] sw_vers
```

```
[RO] uname -m
```

Typische Architekturbezeichnungen

Bezeichnung	Bedeutung
`x86_64` beziehungsweise `AMD64`	64-Bit-x86
`aarch64` beziehungsweise `ARM64`	64-Bit-ARM
`i386` oder `i686`	32-Bit-x86
`arm64` unter macOS	Apple Silicon
`x86_64` unter macOS	Intel-Mac oder gegebenenfalls übersetzter Prozess

> Die Prozessorarchitektur des Hosts, die Architektur des Betriebssystems und die Architektur eines einzelnen Programms können unterschiedlich sein.

4. Windows-Updatequelle und Richtlinien prüfen

Windows-Geräte können Updates aus unterschiedlichen Quellen beziehen:

- Windows Update
- Microsoft Update
- Windows Server Update Services, kurz WSUS

- Microsoft Intune oder andere Geräteverwaltung
- Windows Update for Business
- Lokale oder zentrale Richtlinien
- Manuell bereitgestellte Updatepakete

Windows-Update-Richtlinien anzeigen

```
[RO][SENS] Get-ItemProperty `
    -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate' `
    -ErrorAction SilentlyContinue
```

Richtlinien für automatische Updates anzeigen

```
[RO][SENS] Get-ItemProperty `
    -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU' `
    -ErrorAction SilentlyContinue
```

Angewendete Gruppenrichtlinien dokumentieren

```
[RO][SENS] gpresult /scope computer /r
```

Ausführlichen HTML-Bericht erstellen:

```
[RO][FILE][SENS] gpresult /scope computer /h "<ZIELPFAD>\gpresult-computer.html"
```

Relevante Hinweise

Beobachtung	Mögliche Bedeutung
Interner Update-Server eingetragen	Gerät verwendet wahrscheinlich WSUS
WSUS-Server nicht erreichbar	Erkennung und Download können scheitern
Keine sichtbare lokale Richtlinie	Einstellung kann über MDM oder andere Verwaltung kommen
Einstellung springt zurück	Zentrale Richtlinie überschreibt lokale Änderung
Update wird anderen Geräten angeboten	Zielgruppen-, Freigabe- oder Kompatibilitätsunterschied möglich
Gerät erhält nur bestimmte Updatearten	Richtlinie oder Updatekanal begrenzt das Angebot

> Registrywerte dürfen nicht allein aufgrund einer Internetanleitung gelöscht oder verändert werden. Zuerst muss festgestellt werden, ob sie zentral verwaltet werden.

5. Windows-Update-Dienste untersuchen

Relevante Dienste anzeigen

```
[RO] Get-Service -Name wuauserv, bits, cryptsvc, trustedinstaller |
    Select-Object Name, DisplayName, Status, StartType
```

Bedeutung:

Dienst	Aufgabe
`wuauserv`	Windows Update
`BITS`	Hintergrundübertragung von Dateien
`CryptSvc`	Kryptografische Dienste und Signaturprüfungen
`TrustedInstaller`	Windows Modules Installer

Ausführliche Dienstinformationen

```
[RO] Get-CimInstance Win32_Service |
    Where-Object {
        $_.Name -in 'wuauserv','BITS','CryptSvc','TrustedInstaller'
    } |
    Select-Object Name, State, StartMode, StartName, ExitCode
```

Abhängigkeiten eines Dienstes anzeigen

```
[RO] Get-Service -Name wuauserv -RequiredServices
```

Wichtige Bewertung

- Ein beendeter Dienst ist nicht automatisch fehlerhaft.
- Einige Dienste werden bedarfsgesteuert gestartet.
- Der Starttyp `Manual` kann vorgesehen sein.

- Zentral verwaltete Systeme können abweichende Konfigurationen verwenden.
- Ein Dienstneustart behebt nicht die Ursache einer defekten Richtlinie oder Updatequelle.

6. Installierte Windows-Updates und Verlauf prüfen

Ausgewählte installierte Updates anzeigen

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

> `Get-HotFix` zeigt nicht jede denkbare Update-, Treiber-, App- oder Servicing-Komponente an. Die Ausgabe darf deshalb nicht als vollständiges Inventar aller installierten Aktualisierungen interpretiert werden.

Installierte Windows-Pakete mit DISM anzeigen

```
[RO][PRIV] DISM.exe /Online /Get-Packages /Format:Table
```

Nach einem bestimmten KB-Paket suchen

```
[RO] Get-HotFix -Id <KB-NUMMER> -ErrorAction SilentlyContinue
```

Beispiel:

```
[RO] Get-HotFix -Id KB5030000 -ErrorAction SilentlyContinue
```

Die KB-Nummer im Beispiel ist durch die tatsächlich zu prüfende Nummer zu ersetzen.

Installationszeit und Paketstatus mit DISM prüfen

```
[RO][PRIV] DISM.exe /Online /Get-PackageInfo /PackageName:<PAKETNAME>
```

Den exakten Paketnamen erhält man zuvor über:

```
[RO][PRIV] DISM.exe /Online /Get-Packages /Format:Table
```

7. Windows-Updateprotokolle auswerten

Moderne Windows-Versionen speichern Windows-Update-Ereignisse überwiegend als ETL-Daten. `Get-WindowsUpdateLog` erzeugt daraus eine lesbare Protokolldatei.

Lesbares WindowsUpdate-Protokoll erzeugen

```
[RO][FILE][SENS] Get-WindowsUpdateLog -LogPath "<ZIELPFAD>\WindowsUpdate.log"
```

> Der Befehl sammelt und konvertiert vorhandene Protokolldaten. Die erzeugte Datei kann Servernamen, URLs, Geräteinformationen und Fehlerdetails enthalten.

Windows-Update-Ereignisse anzeigen

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

Nur Warnungen und Fehler der letzten 24 Stunden

```
[RO][SENS] Get-WinEvent -FilterHashtable @{
    LogName   = 'Microsoft-Windows-WindowsUpdateClient/Operational'
    StartTime = (Get-Date).AddHours(-24)
} |
Where-Object {
    $_.LevelDisplayName -in 'Warning','Error','Warnung','Fehler'
} |
Select-Object TimeCreated, LevelDisplayName, Id, Message
```

Servicing-Protokolle

```
[RO][FILE][SENS] %WINDIR%\Logs\CBS\CBS.log
```

```
[RO][FILE][SENS] %WINDIR%\Logs\DISM\dism.log
```

Setup- und Upgradeprotokolle können unter anderem hier liegen

```
[RO][FILE][SENS] %WINDIR%\Panther
```

```
[RO][FILE][SENS] %WINDIR%\Logs\MoSetup
```

Die tatsächlich vorhandenen Dateien hängen von Updateart, Betriebssystemversion und Fehlerphase ab.

Bei der Protokollanalyse suchen nach

- Fehlercode
- KB-Nummer
- Paketname
- Zeitpunkt des Installationsversuchs
- Downloadfehler
- Signaturfehler
- Paketabhängigkeit
- ausstehendem Neustart
- beschädigten Komponenten
- Rollback-Ursache

8. Ausstehenden Windows-Neustart erkennen

Ein ausstehender Neustart kann weitere Updates, Rolleninstallationen oder Paketänderungen blockieren.

Component-Based-Servicing prüfen

```
[RO] Test-Path `
    'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Component Based Servicing\RebootPending'
```

Windows-Update-Neustartstatus prüfen

```
[RO] Test-Path `
    'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\RebootRequired'
```

Ausstehende Dateiumbenennungen prüfen

```
[RO][SENS] Get-ItemProperty `
    -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager' `
    -Name PendingFileRenameOperations `
    -ErrorAction SilentlyContinue
```

Bewertung

- Nicht jeder vorhandene Indikator bedeutet denselben Neustartgrund.
- Ein Neustart muss mit den Protokollen und dem Änderungsverlauf abgeglichen werden.
- Produktivserver dürfen nicht unkontrolliert neu gestartet werden.
- Vor dem Neustart müssen Dienste, Benutzer, Clusterstatus und Wartungsfenster geprüft werden.

9. Windows-Komponentenspeicher prüfen

Beschädigungen im Windows-Komponentenspeicher können Updates und Rolleninstallationen verhindern.

Schnelle Prüfung

```
[TEST][PRIV] DISM.exe /Online /Cleanup-Image /CheckHealth
```

`/CheckHealth` prüft, ob bereits eine Beschädigung markiert wurde. Es führt keinen vollständigen Scan aus.

Ausführlichere Prüfung

```
[TEST][PRIV] DISM.exe /Online /Cleanup-Image /ScanHealth
```

Systemdateien prüfen

```
[TEST][PRIV] sfc /verifyonly
```

Reparatur des Komponentenspeichers

```
[CHANGE][PRIV] DISM.exe /Online /Cleanup-Image /RestoreHealth
```

Systemdateien prüfen und reparieren

```
[CHANGE][PRIV] sfc /scannow
```

Wichtige Hinweise

- `RestoreHealth` kann Dateien von Windows Update beziehen.
- In abgeschotteten Umgebungen kann eine passende Reparaturquelle erforderlich sein.
- Eine Reparaturquelle muss zur Betriebssystemversion und zum Wartungsstand passen.
- DISM und SFC können längere Zeit benötigen.
- Ein Abbruch sollte nicht allein wegen eines länger unveränderten Prozentwerts erfolgen.
- Protokolle befinden sich unter anderem in `CBS.log` und `dism.log`.

> DISM und SFC sind keine pauschalen ersten Schritte für jeden Updatefehler. Zuerst müssen Fehlercode, Quelle, Speicherplatz und Updatephase geprüft werden.

10. Windows-Proxy und Netzwerkpfad prüfen

Windows-Systemdienste können andere Proxyeinstellungen verwenden als ein angemeldeter Benutzer.

WinHTTP-Proxy anzeigen

```
[RO] netsh winhttp show proxy
```

DNS-Auflösung prüfen

```
[RO] Resolve-DnsName "<UPDATE-ODER-WSUS-SERVER>"
```

HTTPS-Erreichbarkeit eines konkret bekannten Servers prüfen

```
[TEST] Test-NetConnection `
    -ComputerName "<SERVERNAME>" `
    -Port 443
```

Bei einem internen WSUS-Server muss der tatsächlich konfigurierte Port verwendet werden.

Route anzeigen

```
[RO] Test-NetConnection `
    -ComputerName "<SERVERNAME>" `
    -TraceRoute
```

Typische Ursachen

- Proxy verlangt eine interaktive Anmeldung.
- WinHTTP-Proxy ist veraltet.
- PAC-Datei funktioniert nur im Benutzerkontext.
- Firewall blockiert Updateendpunkte.
- TLS-Inspection verändert Zertifikate.
- DNS liefert eine falsche oder interne Adresse.
- VPN verändert Route oder Namensauflösung.
- WSUS ist erreichbar, aber dessen Datenbank oder Anwendungspool ist gestört.

> Öffentliche Updateplattformen verwenden mehrere dynamische Endpunkte. Es sollten keine unbestätigten Einzeladressen dauerhaft freigeschaltet werden; maßgeblich ist die aktuelle Herstellerdokumentation.

11. Debian- und Ubuntu-Paketverwaltung prüfen

Debian und Ubuntu verwenden üblicherweise APT und dpkg.

Betriebssystemversion prüfen

```
[RO][FILE] cat /etc/os-release
```

Konfigurierte Paketquellen anzeigen

```
[RO][FILE][SENS] grep -RhsE '^[[:space:]]*deb ' \
    /etc/apt/sources.list /etc/apt/sources.list.d 2>/dev/null
```

Neuere APT-Quellen können zusätzlich im Deb822-Format mit der Endung `.sources` gespeichert sein:

```
[RO][FILE][SENS] grep -RhsE \
    '^[[:space:]]*(Types|URIs|Suites|Components|Signed-By):' \
    /etc/apt/sources.list.d 2>/dev/null
```

Paketstatus auf Inkonsistenzen prüfen

```
[RO][PRIV] sudo dpkg --audit
```

Alternativ:

```
[RO][PRIV] sudo dpkg -C
```

Paketabhängigkeiten prüfen

```
[TEST][PRIV] sudo apt-get check
```

Installierte und verfügbare Version eines Pakets anzeigen

```
[RO] apt-cache policy "<PAKETNAME>"
```

Zurückgehaltene Pakete anzeigen

```
[RO] apt-mark showhold
```

Paketarchitektur anzeigen

```
[RO] dpkg --print-architecture
```

Zusätzlich konfigurierte Architekturen:

```
[RO] dpkg --print-foreign-architectures
```

Letzte dpkg-Aktionen anzeigen

```
[RO][FILE][SENS] tail -n 100 /var/log/dpkg.log
```

APT-Verlauf anzeigen, sofern vorhanden

```
[RO][FILE][SENS] ls -l /var/log/apt
```

```
[RO][FILE][SENS] tail -n 100 /var/log/apt/history.log
```

12. APT-Aktualisierung und Fehlerausgabe kontrolliert prüfen

Das Aktualisieren der Paketlisten ist bereits eine Veränderung des lokalen Paketcache.

```
[CHANGE][PRIV] sudo apt update
```

Dabei sind besonders folgende Meldungen zu beachten:

Meldung	Mögliche Ursache
`Temporary failure resolving`	DNS-Problem
`Connection timed out`	Netzwerk, Proxy, Firewall oder Repository
`404 Not Found`	Falscher Distributionsname oder veraltete Quelle
`NO_PUBKEY`	Fehlender oder falsch eingebundener Signaturschlüssel
`Release file is not valid yet`	Systemzeit möglicherweise falsch
`does not have a Release file`	Quelle unterstützt die Distribution nicht
`Hash Sum mismatch`	Spiegelserver, Proxycache oder inkonsistente Metadaten
`Could not get lock`	Andere Paketverwaltung ist aktiv
`Unmet dependencies`	Abhängigkeitskonflikt
`held broken packages`	Zurückgehaltene oder nicht auflösbare Pakete
`No space left on device`	Speicherplatz oder Inodes ausgeschöpft

Simulation einer Installation

```
[TEST] apt-get --simulate install "<PAKETNAME>"
```

Die Simulation verändert keine Pakete, bildet aber nicht jede Nebenwirkung eines tatsächlichen Installationsskripts ab.

Paketinformationen anzeigen

```
[RO] apt-cache show "<PAKETNAME>"
```

Abhängigkeiten anzeigen

```
[RO] apt-cache depends "<PAKETNAME>"
```

Reverse-Abhängigkeiten anzeigen

```
[RO] apt-cache rdepends "<PAKETNAME>"
```

13. Paketmanager-Sperren unter Linux untersuchen

Eine Sperrdatei darf nicht allein aufgrund ihrer Existenz gelöscht werden. Zuerst muss geprüft werden, ob ein Paketmanager tatsächlich arbeitet.

APT- und dpkg-Prozesse anzeigen

```
[RO][SENS] ps -ef |
    grep -E '[a]pt|[d]pkg|[u]nattended-upgrade'
```

Prozesse mit geöffneten APT- oder dpkg-Sperren anzeigen

```
[RO][PRIV][SENS] sudo lsof \
    /var/lib/dpkg/lock \
    /var/lib/dpkg/lock-frontend \
    /var/lib/apt/lists/lock \
    /var/cache/apt/archives/lock 2>/dev/null
```

Automatische Updates prüfen

```
[RO] systemctl status apt-daily.service --no-pager
```

```
[RO] systemctl status apt-daily-upgrade.service --no-pager
```

Protokolle automatischer Updates

```
[RO][PRIV][SENS] sudo journalctl \
    -u apt-daily.service \
    -u apt-daily-upgrade.service \
    --since "-24 hours" --no-pager
```

> Das Löschen einer Sperrdatei oder Beenden eines Paketmanagers während einer Installation kann die Paketdatenbank beschädigen. Erst muss geklärt werden, ob der Prozess noch arbeitet, wartet oder abgestürzt ist.

14. Unvollständige dpkg-Transaktionen behandeln

Ein abgebrochener Installationsprozess kann Pakete in einem nicht fertig konfigurierten Zustand hinterlassen.

Zustand zunächst prüfen

```
[RO][PRIV] sudo dpkg --audit
```

Paketstatus eines bestimmten Pakets anzeigen

```
[RO] dpkg-query -W \
    -f='${Package}\t${Status}\t${Version}\n' "<PAKETNAME>"
```

Noch nicht konfigurierte Pakete konfigurieren

```
[CHANGE][PRIV] sudo dpkg --configure -a
```

Abhängigkeiten reparieren

```
[CHANGE][PRIV] sudo apt-get --fix-broken install
```

Diese Befehle können:

- Paketkonfigurationen ausführen
- Dienste starten oder neu starten
- Konfigurationsdateien verändern
- zusätzliche Pakete installieren
- Pakete entfernen
- Administratorfragen anzeigen

Vorher sollten geprüft werden:

- Sicherung oder Snapshot
- freier Speicherplatz
- aktueller Paketstatus
- betroffene Produktivdienste
- vorgeschlagene Paketänderungen
- Wartungsfenster

> Bei `apt-get --fix-broken install` muss die vorgeschlagene Paketliste vor der Bestätigung vollständig gelesen werden.

15. DNF- und RPM-Systeme untersuchen

Fedora, Red Hat Enterprise Linux und verwandte Distributionen verwenden häufig DNF und RPM.

Betriebssystemversion

```
[RO][FILE] cat /etc/os-release
```

Aktive Repositories anzeigen

```
[RO][SENS] dnf repolist
```

Alle bekannten Repositories anzeigen

```
[RO][SENS] dnf repolist --all
```

Paketdatenbank und Abhängigkeiten prüfen

```
[TEST][PRIV] sudo dnf check
```

Verfügbare Aktualisierungen prüfen

```
[TEST][PRIV] sudo dnf check-update
```

> `dnf check-update` kann mit Rückgabecode `100` enden, wenn Updates verfügbar sind. Dieser Rückgabecode bedeutet dann nicht automatisch einen Fehler.

Paketinformationen anzeigen

```
[RO] dnf info "<PAKETNAME>"
```

Installierte Paketversion anzeigen

```
[RO] rpm -q "<PAKETNAME>"
```

Paketarchitektur anzeigen

```
[RO] rpm -q --queryformat \
   '%{NAME}\t%{VERSION}-%{RELEASE}\t%{ARCH}\n' "<PAKETNAME>"
```

Transaktionsverlauf anzeigen

```
[RO][SENS] dnf history
```

Details einer Transaktion

```
[RO][SENS] dnf history info <TRANSAKTIONS-ID>
```

Installierte Dateien gegen RPM-Metadaten prüfen

```
[TEST][PRIV][FILE][SENS] sudo rpm -Va
```

`rpm -Va` kann lange laufen und meldet auch absichtlich veränderte Konfigurationsdateien. Abweichungen sind nicht automatisch Beschädigungen.

16. Zypper- und RPM-Systeme untersuchen

SUSE Linux Enterprise und openSUSE verwenden typischerweise Zypper.

Repositories anzeigen

```
[RO][SENS] zypper repos
```

Ausführlicher:

```
[RO][SENS] zypper repos --details
```

Verfügbare Patches anzeigen

```
[TEST][PRIV] sudo zypper list-patches
```

Abhängigkeiten prüfen

```
[TEST][PRIV] sudo zypper verify
```

Paketinformationen anzeigen

```
[RO] zypper info "<PAKETNAME>"
```

Installierte Version prüfen

```
[RO] rpm -q "<PAKETNAME>"
```

Zypper-Protokoll lesen

```
[RO][FILE][SENS] tail -n 100 /var/log/zypper.log
```

17. macOS-Softwareupdates untersuchen

macOS-Version und Build anzeigen

```
[RO] sw_vers
```

Verfügbare Apple-Softwareupdates suchen

```
[TEST] softwareupdate --list
```

Der Befehl kontaktiert die konfigurierte Updateinfrastruktur und kann einige Zeit benötigen.

Verlauf installierter Apple-Updates anzeigen

```
[RO] softwareupdate --history
```

Ausführlicher Verlauf:

```
[RO] softwareupdate --history --all
```

Hilfe der installierten softwareupdate-Version anzeigen

```
[RO] softwareupdate --help
```

> Die verfügbaren Optionen von `softwareupdate` können sich zwischen macOS-Versionen unterscheiden. Deshalb sollte die lokale Hilfe geprüft werden, bevor Installationsoptionen verwendet werden.

Freien Speicherplatz prüfen

```
[RO] df -h /
```

Datum und Uhrzeit prüfen

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

Netzwerkzeitstatus prüfen

```
[RO][PRIV] sudo systemsetup -getusingnetworktime
```

Eingetragenen Zeitserver anzeigen

```
[RO][PRIV][SENS] sudo systemsetup -getnetworktimeserver
```

MDM-Registrierungsstatus anzeigen

```
[RO][PRIV][SENS] sudo profiles status -type enrollment
```

Ein verwalteter Mac kann Updates aufgrund von MDM-Vorgaben:

- verzögert angeboten bekommen
- zu einem Termin erzwungen installieren
- nur über bestimmte Updatekanäle erhalten
- mit anderen Neustartbedingungen installieren
- nicht manuell verändern dürfen

18. macOS-Updateprotokolle auswerten

Softwareupdate-Protokolle der letzten Stunde

```
[RO][PRIV][SENS] sudo log show --last 1h \
    --predicate 'process == "softwareupdated"' \
    --style compact
```

Mehrere typische Updateprozesse berücksichtigen

```
[RO][PRIV][SENS] sudo log show --last 2h \
    --predicate 'process == "softwareupdated" OR process == "softwareupdate"' \
    --style compact
```

Nur Fehler und schwerwiegende Meldungen

```
[RO][PRIV][SENS] sudo log show --last 2h \
    --predicate '(process == "softwareupdated" OR process == "softwareupdate") AND messageType >= error' \
    --style compact
```

Die von macOS verwendeten Prozesse und Protokollfelder können sich zwischen Versionen unterscheiden. Wenn die Abfrage keine Ergebnisse liefert, bedeutet dies nicht automatisch, dass kein Updateversuch stattgefunden hat.

Typische Ursachen

- Zu wenig Speicherplatz
- Nicht unterstütztes Mac-Modell
- Falsche Systemzeit
- Netzwerk- oder CDN-Problem
- MDM-Aufschub
- Fehlende Benutzerbestätigung
- FileVault- oder Bootstrap-Token-Anforderung in verwalteten Umgebungen
- Beschädigter Download
- Nicht abgeschlossener vorheriger Updateversuch
- Firmware- oder Wiederherstellungsfehler
- Update wird für den verwendeten Kanal noch nicht angeboten

19. Repository-, Signatur- und Zertifikatsfehler unterscheiden

Softwarepakete und Repositorymetadaten werden normalerweise kryptografisch geprüft.

Typische Fehlerursachen

Ursache	Wirkung
Falsche Systemzeit	Zertifikat oder signierte Metadaten erscheinen ungültig
Abgelaufener Signaturschlüssel	Repository wird nicht akzeptiert
Falscher Schlüssel	Signatur kann nicht verifiziert werden

Ursache	Wirkung
Manipulierender Proxy	TLS- oder Zertifikatsprüfung schlägt fehl
Unvollständiger Download	Prüfsumme stimmt nicht
Veraltete Repositorydefinition	Release-Datei oder Paketpfad fehlt
Nicht unterstützte Distribution	Repository stellt keine passenden Metadaten bereit
Falsche Architektur	Paket kann nicht installiert werden
Internes Zertifikat fehlt	Unternehmensrepository wird nicht vertraut
Spiegelserver nicht synchron	Metadaten und Paketdateien passen vorübergehend nicht zusammen

Sicherheitsregel

Signatur- und Zertifikatsprüfungen dürfen nicht dauerhaft deaktiviert werden, um eine Installation „zum Laufen zu bringen“.

Stattdessen müssen geprüft werden:

1. Systemzeit
2. Repositoryadresse
3. Distributionsversion
4. Paketarchitektur
5. erwarteter Signaturschlüssel
6. Zertifikatskette
7. Proxy- oder TLS-Inspection
8. Integrität des Downloads
9. offizielle Installationsanleitung des Herstellers

20. Abhängigkeiten und Versionskonflikte analysieren

Paketmanager installieren selten nur eine einzelne Datei. Pakete besitzen Abhängigkeiten, Konflikte und Versionsanforderungen.

Typische Beziehungen

Beziehung	Bedeutung
Abhängigkeit	Zusätzlich benötigtes Paket
Versionsabhängigkeit	Nur bestimmte Versionen sind zulässig
Konflikt	Pakete dürfen nicht gleichzeitig installiert sein
Ersatz	Ein Paket ersetzt ein anderes

Beziehung	Bedeutung
Empfehlung	Zusätzliches Paket wird empfohlen
Architekturabhängigkeit	Paket benötigt eine bestimmte Plattform
Modulschnittstelle	Anwendung benötigt eine kompatible Bibliotheksversion

APT-Abhängigkeiten anzeigen

```
[RO] apt-cache depends "<PAKETNAME>"
```

APT-Reverse-Abhängigkeiten

```
[RO] apt-cache rdepends "<PAKETNAME>"
```

RPM-Abhängigkeiten anzeigen

```
[RO] rpm -qR "<PAKETNAME>"
```

Paket ermitteln, das eine Fähigkeit bereitstellt

DNF:

```
[RO] dnf provides "<DATEI-ODER-FÄHIGKEIT>"
```

APT:

```
[RO] apt-file search "<DATEINAME>"
```

`apt-file` ist nicht standardmäßig auf jedem System installiert und benötigt einen eigenen Suchindex.

Diagnosefragen

- Ist das benötigte Repository aktiviert?
- Ist ein Paket bewusst festgehalten worden?
- Wurden Fremdquellen gemischt?
- Wird ein Paket aus einer anderen Distributionsversion verwendet?
- Verlangt die Anwendung eine ältere Bibliothek?
- Wurde ein Teilupdate durchgeführt?

- Passt die Architektur?
- Ist ein Herstellerrepository noch unterstützt?

21. Paketcache und heruntergeladene Dateien richtig bewerten

Beschädigte oder unvollständige Downloads können wiederholte Fehler verursachen. Das pauschale Löschen aller Updatecaches ist dennoch kein sinnvoller erster Diagnoseschritt.

Vor einer Cachebereinigung prüfen

- Ist der Fehler reproduzierbar?
- Ist die Prüfsumme falsch?
- Betrifft der Fehler immer dieselbe Datei?
- Liegt ein Proxycache zwischen Client und Repository?
- Ist genügend Speicherplatz vorhanden?
- Gibt es noch einen aktiven Paketmanager?
- Werden Dateien für einen Rollback benötigt?
- Kann der Download aus einer vertrauenswürdigen Quelle erneut erfolgen?

APT-Downloadcache anzeigen

```
[RO][FILE] du -sh /var/cache/apt/archives
```

DNF-Cachebelegung anzeigen

```
[RO][FILE][PRIV] sudo du -sh /var/cache/dnf
```

Windows-Update-Downloadbereich

```
[RO][FILE][PRIV] %WINDIR%\SoftwareDistribution\Download
```

> Der Ordner `SoftwareDistribution` darf nicht unkontrolliert umbenannt oder gelöscht werden. Ein Reset der Windows-Updatekomponenten verändert den Updatezustand und erschwert möglicherweise die ursprüngliche Ursachenanalyse.

22. Automatische Updates und Wartungsfenster prüfen

Automatische Aktualisierungen können parallel zu manuellen Arbeiten laufen oder durch Wartungsregeln eingeschränkt sein.

Windows

Zu prüfen sind:

- Gruppenrichtlinien
- Windows Update for Business
- WSUS-Zielgruppen
- aktive Nutzungszeit
- Neustartrichtlinien
- Intune-Update-Ringe
- geplante Wartungsfenster

Debian und Ubuntu

```
[RO] systemctl status apt-daily.timer --no-pager
```

```
[RO] systemctl status apt-daily-upgrade.timer --no-pager
```

```
[RO][FILE] systemctl list-timers --all |
    grep -E 'apt|upgrade'
```

DNF-Systeme

```
[RO] systemctl list-timers --all |
    grep -E 'dnf|upgrade'
```

Je nach Distribution und Konfiguration kann beispielsweise `dnf-automatic` verwendet werden.

macOS

Zu prüfen sind:

- Einstellungen für automatische Updates
- MDM-Aufschübe
- erzwungene Installationstermine
- erforderliche Benutzeraktion
- Neustartbedingungen
- verfügbare Bootstrap- oder Secure Tokens

> Automatische Updateprozesse dürfen nicht beendet werden, ohne ihren aktuellen Transaktionszustand zu prüfen.

23. Updatefehler nach dem Neustart analysieren

Ein Update kann zunächst erfolgreich erscheinen und erst beim Neustart scheitern.

Typische Symptome

- Endlose Neustartschleife
- „Änderungen werden rückgängig gemacht“
- Dienst startet nicht
- Treiber wird nicht geladen
- Netzwerkverbindung fehlt
- Anwendung ist inkompatibel
- Kernel startet nicht
- Dateisystem muss geprüft werden
- Firmwareaktualisierung schlägt fehl
- Verschlüsseltes Systemvolume wird nicht entsperrt

Prüfreihenfolge

1. Exakte Meldung und Zeitpunkt dokumentieren.
2. Prüfen, ob ein automatischer Rollback erfolgt ist.
3. Update- und Bootprotokolle sichern.
4. Zuletzt installierte Pakete ermitteln.
5. Dienst- und Treiberstatus kontrollieren.
6. Freien Speicherplatz prüfen.
7. Abhängige Systeme und Clusterstatus berücksichtigen.
8. Herstellerhinweise zum konkreten Update prüfen.
9. Rücknahme nur nach Bewertung der Sicherheitsauswirkung durchführen.
10. Vor erneuter Installation die ursprüngliche Ursache beheben.

Linux: vorherigen Boot untersuchen

```
[RO][PRIV][SENS] sudo journalctl -b -1 -p warning --no-pager
```

Windows: letzte Systemfehler

```
[RO][SENS] Get-WinEvent -FilterHashtable @{
    LogName   = 'System'
    StartTime = (Get-Date).AddHours(-24)
} |
Where-Object {
    $_.LevelDisplayName -in 'Warning','Error','Warnung','Fehler'
} |
Select-Object TimeCreated, ProviderName, Id, LevelDisplayName, Message
```

24. Updates in Clustern und hochverfügbaren Systemen

Bei Clustern, Replikationssystemen und redundanten Diensten reicht die Betrachtung eines einzelnen Servers nicht aus.

Vor einem Update prüfen

- Ist der Cluster gesund?
- Funktioniert die Replikation?
- Ist ausreichend Redundanz vorhanden?
- Welcher Knoten besitzt aktuell die produktiven Rollen?
- Können Rollen kontrolliert verschoben werden?
- Ist Quorum beziehungsweise Witness erreichbar?
- Sind alle Knoten kompatibel?
- Unterstützt der Hersteller gemischte Patchstände?
- Existiert ein getesteter Rückfallplan?
- Ist das Monitoring aktiv?
- Wurde eine Sicherung geprüft?

Gefahren

- Mehrere Knoten werden gleichzeitig neu gestartet.
- Ein bereits gestörter Knoten wird als Ausweichknoten eingeplant.
- Datenbankschema und Anwendungsversion passen nicht zusammen.
- Quorum geht verloren.
- Load Balancer leitet weiterhin auf einen gewarteten Knoten.
- Replikation wird durch Versionsunterschiede unterbrochen.
- Ein automatisches Update umgeht das Wartungsfenster.

> Ein Snapshot eines einzelnen Clusterknotens ist kein vollständiger Rückfallplan für einen verteilten Dienst.

25. Sichere Reihenfolge der Updatefehleranalyse

1. Betroffenes Update, Paket oder KB eindeutig bestimmen.
2. Originalen Fehlercode und Fehlerzeitpunkt dokumentieren.
3. Betriebssystemversion, Build und Architektur prüfen.
4. Updatephase bestimmen.
5. Freien Speicherplatz und Inodes kontrollieren.
6. Datum, Uhrzeit und Zeitzone prüfen.
7. Ausstehenden Neustart berücksichtigen.
8. Updatequelle, Repository, WSUS oder MDM ermitteln.
9. DNS, Proxy, Firewall und Erreichbarkeit prüfen.
10. Paketmanager- beziehungsweise Updatedienste untersuchen.
11. Protokolle zum Fehlerzeitpunkt auswerten.
12. Paketabhängigkeiten und Signaturen prüfen.
13. Aktive Installations- oder Sperrprozesse identifizieren.
14. Systemdateien beziehungsweise Paketdatenbank kontrollieren.
15. Erst danach Cachebereinigung oder Reparaturmaßnahmen planen.
16. Änderung im Wartungsfenster durchführen.
17. Neustartbedarf kontrolliert umsetzen.
18. Update-, Dienst- und Anwendungsfunktion verifizieren.
19. Ergebnis und verbleibende Risiken dokumentieren.

26. Typische Fehlinterpretationen vermeiden

Fehlinterpretation	Richtige Bewertung
„Der Download hängt bei 0 Prozent.“	Verbindung, Metadatenprüfung oder Vorbereitung kann bereits laufen.
„Der Update-Dienst ist beendet.“	Bedarfsgesteuerter Dienst kann regulär beendet sein.
„Ping funktioniert, also ist Windows Update erreichbar.“	DNS, HTTPS, Proxy, CDN und Authentifizierung sind zusätzlich relevant.
„Ein Neustart ist immer harmlos.“	Produktivdienste, Cluster und Benutzer müssen berücksichtigt werden.
„Cache löschen behebt jeden Updatefehler.“	Es entfernt Symptome und erschwert möglicherweise die Diagnose.
„Signaturprüfung kann vorübergehend deaktiviert werden.“	Dadurch wird eine wichtige Sicherheitskontrolle umgangen.

Fehlinterpretation	Richtige Bewertung
„Get-HotFix zeigt alle installierten Updates.“	Es erfasst nicht jede Update- und Paketart.
„Rückgabecode 100 von dnf ist ein Fehler.“	Bei `dnf check-update` kann er verfügbare Updates anzeigen.
„Eine Sperrdatei kann einfach gelöscht werden.“	Eine aktive Pakettransaktion könnte beschädigt werden.
„Update fehlgeschlagen bedeutet fehlerhaftes Update.“	Ursache kann lokal in Speicher, Netzwerk, Zeit oder Paketstatus liegen.
„Snapshot entspricht einer Sicherung.“	Snapshot und unabhängige Sicherung erfüllen unterschiedliche Aufgaben.
„Update erfolgreich bedeutet Dienst funktionsfähig.“	Nachkontrolle von Diensten und Anwendungen bleibt erforderlich.

27. Schnelle Befehlsübersicht

Aufgabe	Windows	Debian/Ubuntu	RHEL/Fedora	macOS
Betriebssystemversion	`[RO] Get-ComputerInfo`	`[RO] cat /etc/os-release`	`[RO] cat /etc/os-release`	`[RO] sw_vers`
Architektur	`[RO] Get-CimInstance Win32_OperatingSystem`	`[RO] dpkg --print-architecture`	`[RO] uname -m`	`[RO] uname -m`
Freier Speicher	`[RO] Get-Volume`	`[RO] df -hT`	`[RO] df -hT`	`[RO] df -h`
Updatequelle	Richtlinien und WSUS prüfen	APT-Quelldateien prüfen	`[RO] dnf repolist`	MDM und Updateeinstellungen
Updatedienste	`[RO] Get-Service wuauserv,bits,cryptsvc,trustedinstaller`	APT-Timer und Prozesse	DNF-Timer und Prozesse	macOS-Systemdienst
Updatehistorie	`Get-HotFix` und DISM	APT- und dpkg-Protokolle	`[RO] dnf history`	`[RO] softwareupdate --history`
Verfügbare Updates	Windows-Updateverwaltung	`[CHANGE][PRIV] sudo apt update`	`[TEST][PRIV] sudo dnf check-update`	`[TEST] softwareupdate --list`
Paketstatus prüfen	DISM	`[RO][PRIV] sudo dpkg --audit`	`[TEST][PRIV] sudo dnf check`	Updateprotokoll prüfen
Abhängigkeiten prüfen	CBS- und DISM-Protokoll	`[TEST][PRIV] sudo apt-get check`	`[TEST][PRIV] sudo dnf check`	Nicht direkt vergleichbar
Paketversion	DISM oder `Get-HotFix`	`[RO] apt-cache policy "<PAKET>"`	`[RO] rpm -q "<PAKET>"`	Anwendungsspezifisch
Ausstehender Neustart	Registryindikatoren prüfen	`/var/run/reboot-required` kann distributionsabhängig existieren	Paket- und Kernelstatus prüfen	Update- und MDM-Status prüfen

Aufgabe	Windows	Debian/Ubuntu	RHEL/Fedora	macOS
Protokolle	`Get-WindowsUpdateLog`, CBS, Ereignisanzeige	`/var/log/apt`, `/var/log/dpkg.log`, Journal	`dnf history`, Journal	`[RO][PRIV] log show ...`
Systemintegrität	`[TEST][PRIV] DISM /ScanHealth`	Paketdatenbank prüfen	`[TEST][PRIV] rpm -Va`	Festplatten- und Updateprotokolle
Proxy	`[RO] netsh winhttp show proxy`	APT- und Umgebungsvariablen prüfen	DNF- und Umgebungsvariablen prüfen	Netzwerk- und MDM-Konfiguration

28. Entscheidungsmatrix

Befund	Wahrscheinliche Ursache	Nächster Schritt
Update wird nicht angeboten	Richtlinie, Kanal, Kompatibilität oder Supportstatus	Quelle und Voraussetzungen prüfen
Download startet nicht	DNS, Proxy, Firewall oder Repository	Netzwerkpfad untersuchen
Download wiederholt beschädigt	Proxycache, Spiegelserver oder Datenträger	Prüfsumme, Quelle und Speicher prüfen
Signatur ungültig	Zeit, Schlüssel oder Zertifikatskette	Zeit und Vertrauenskette prüfen
Nicht genügend Speicher	System- oder temporäres Volume voll	Speicherverbrauch analysieren
Paketmanager ist gesperrt	Andere Transaktion aktiv	Prozess und Protokolle prüfen
Paketabhängigkeit nicht erfüllbar	Fehlendes Repository oder Versionskonflikt	Abhängigkeitsbaum untersuchen
Neustart wird wiederholt verlangt	Ausstehende Operation nicht abgeschlossen	Pending-Reboot-Status und Protokolle prüfen
Update rollt zurück	Installations-, Treiber- oder Bootfehler	Setup- und Systemprotokolle auswerten
Mehrere Geräte betroffen	Zentrale Quelle, Richtlinie oder Herstellerproblem	Gemeinsamen Nenner ermitteln
Nur ein Gerät betroffen	Lokaler Cache, Zustand oder Konfiguration	Mit funktionierendem Vergleichssystem abgleichen
Update erfolgreich, Dienst ausgefallen	Inkompatibilität oder Konfigurationsmigration	Dienstprotokolle und Herstellerhinweise prüfen
Betriebssystem nicht mehr unterstützt	Keine reguläre Updateversorgung	Geplantes Upgrade beziehungsweise Migration
DNF endet mit Code 100	Updates verfügbar	Nicht als Paketmanagerfehler behandeln

29. Dokumentationsvorlage

Ticket:

Datum und Uhrzeit:

Bearbeiter:

Betroffenes System:

Betriebssystem:

Version und Build:

Architektur:

Systemrolle:

Physisch oder virtuell:

Cluster- oder Domänenmitglied:

Updateart:

KB-, Paket- oder Versionsnummer:

Updatequelle:

WSUS, Repository oder MDM:

Fehlerphase:

Originaler Fehlercode:

Originale Fehlermeldung:

Zeitpunkt des Fehlers:

Freier Speicher:

Inode-Belegung:

Datum und Uhrzeit:

Ausstehender Neustart:

Updatedienst:

Paketmanager:

Aktive Installationsprozesse:

Sperrstatus:

DNS-Auflösung:

Proxy:

Erreichbarkeit der Updatequelle:

Signatur- oder Zertifikatsfehler:

Repositorystatus:

Abhängigkeitsfehler:

```
Relevante Protokolle:
Letzte erfolgreiche Aktualisierung:
Letzte Änderungen:
Vergleich mit funktionierendem System:

Vermutete Ursache:
Durchgeführte Tests:
Sicherung geprüft:
Wartungsfenster:
Durchgeführte Änderung:
Neustart durchgeführt:
Rückfallplan:

Installationsstatus nach der Maßnahme:
Dienststatus:
Anwendungsprüfung:
Monitoring unauffällig:
Weiterführende Maßnahmen:
```

Merksatz

> Bei Updatefehlern wird nicht sofort der Cache gelöscht oder das Paket erneut installiert. Zuerst werden Updatequelle, Fehlerphase, Fehlercode, Speicherplatz, Zeit, Netzwerk, Paketstatus und Protokolle geprüft.

Quellen und weiterführende Dokumentation

- Microsoft Learn – Windows-Updateprobleme behandeln
- Microsoft Learn – Windows-Update-Beschädigungen und Installationsfehler beheben
- Microsoft Learn – Get-WindowsUpdateLog
- Microsoft Learn – DISM-Befehlszeilenoptionen
- Microsoft Learn – Windows Update for Business
- Debian-Handbuch – APT
- Ubuntu Server – Paketverwaltung
- Ubuntu Manpages – apt-get

- DNF-Dokumentation – Command Reference
- Red Hat – DNF-Paketverwaltung
- openSUSE – Zypper-Dokumentation
- Apple – Softwareupdate-Prozess überprüfen
- Apple – Softwareupdates mit der Geräteverwaltung anwenden
- Apple-Handbuch – softwareupdate