5.1 Windows – Integrierte Werkzeuge zur Fehleranalyse

Windows stellt zahlreiche grafische Werkzeuge, Befehlszeilenprogramme und PowerShell-Cmdlets zur systematischen Fehleranalyse bereit. Damit lassen sich unter anderem Ereignisprotokolle, Prozesse, Dienste, Ressourcen, Datenträger, Netzwerkverbindungen, Systemdateien und Abstürze untersuchen.

Die Werkzeuge liefern jedoch zunächst nur Befunde. Einzelne Warnungen, hohe Messwerte oder Ereignis-IDs beweisen noch keine Ursache. Ergebnisse müssen immer mit Fehlerzeitpunkt, Symptomen, Änderungen und weiteren Datenquellen abgeglichen werden.

Kennzeichnung der Befehle

Kennzeichnung	Bedeutung
`[RO]`	Nur lesender Befehl
`[TEST]`	Führt einen aktiven Test aus
`[CHANGE]`	Kann den Systemzustand verändern
`[PRIV]`	Erfordert möglicherweise administrative Rechte
`[SENS]`	Ausgabe kann sensible Informationen enthalten
`[RESTART]`	Neustart oder Unterbrechung möglich

> Befehle mit `[CHANGE]`, `[RESTART]` oder `[SENS]` müssen vor ihrer Verwendung besonders sorgfältig geprüft werden.

---

1. Diagnosewerkzeuge gezielt auswählen

Fragestellung	Geeignetes Werkzeug
Was geschah zum Fehlerzeitpunkt?	Ereignisanzeige, `Get-WinEvent`
Welcher Prozess belastet das System?	Task-Manager, Ressourcenmonitor, `Get-Process`
Welcher Dienst läuft nicht?	Diensteverwaltung, `Get-Service`, `sc.exe`
Welches Programm öffnet einen Port?	Ressourcenmonitor, `Get-NetTCPConnection`, `netstat`
Funktioniert die Namensauflösung?	`Resolve-DnsName`, `nslookup`

Fragestellung	Geeignetes Werkzeug
Ist ein TCP-Port erreichbar?	`Test-NetConnection`
Welche Route nimmt ein Paket?	`tracert` `pathping`
Ist ein Datenträger voll?	Explorer, Datenträgerverwaltung, `Get-Volume`
Sind Windows-Systemdateien beschädigt?	SFC und DISM
Warum wurde Windows neu gestartet?	Ereignisanzeige, `Get-WinEvent`
Welche Treiber und Geräte sind vorhanden?	Geräte-Manager, `Get-PnpDevice`, `driverquery`
Wie entwickelt sich die Leistung über Zeit?	Leistungsüberwachung, Datenkollektorsätze
Welche Änderungen gingen der Störung voraus?	Zuverlässigkeitsverlauf, Updateverlauf, Ereignisprotokolle
Warum ist Windows abgestürzt?	Speicherabbild, WinDbg, Ereignisprotokolle

Grundregel

1. Symptom und Zeitpunkt festhalten.
2. Umfang der Störung bestimmen.
3. Ereignisprotokolle zum betreffenden Zeitraum prüfen.
4. Ressourcen und betroffene Komponenten untersuchen.
5. Abhängigkeiten kontrollieren.
6. Erst danach Änderungen durchführen.
7. Ergebnis und Rückfallmöglichkeit dokumentieren.

2. Windows-Version und Systeminformationen erfassen

Windows-Versionsdialog öffnen

```
[RO] winver
```

Systeminformationen anzeigen

```
[RO][SENS] systeminfo
```

Die Ausgabe kann unter anderem Computername, Betriebssystemversion, Installationsdatum, Arbeitsspeicher, Domänenzugehörigkeit und installierte Hotfixes enthalten.

Windows-Version mit PowerShell

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

Betriebssystem über CIM abfragen

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

Computer- und Hardwareinformationen

```
[RO][SENS] Get-ComputerInfo
```

Systeminformationen grafisch öffnen

```
[RO] msinfo32
```

`msinfo32` enthält eine umfassende Übersicht über Hardware, Komponenten, Treiber, Ressourcen und die Softwareumgebung.

> Systeminformationen können Computername, Benutzerbezug, Netzwerkdaten und installierte Software offenlegen. Exporte müssen vor einer Weitergabe geprüft werden.

3. Ereignisanzeige verwenden

Ereignisanzeige öffnen

[RO] eventvwr.msc

Wichtige Windows-Protokolle

Protokoll	Typische Inhalte
Anwendung	Ereignisse von Programmen und Anwendungsdiensten
Sicherheit	Anmeldungen, Zugriffsprüfungen und Sicherheitsereignisse
Setup	Installation und Einrichtung von Windows-Komponenten
System	Treiber, Dienste, Hardware und Betriebssystemkomponenten
Weitergeleitete Ereignisse	Von anderen Computern empfangene Ereignisse

Unter **Anwendungs- und Dienstprotokolle** befinden sich zusätzliche komponentenspezifische Protokolle, beispielsweise für PowerShell, Windows Defender, DNS, Gruppenrichtlinien oder Remotedesktopdienste.

Ereignisstufen

Stufe	Bedeutung
Kritisch	Schwerwiegendes Ereignis, häufig mit Ausfall oder unerwartetem Neustart
Fehler	Eine Funktion oder Operation ist fehlgeschlagen
Warnung	Mögliches Problem oder auffälliger Zustand
Information	Regulärer Vorgang oder Statusmeldung
Ausführlich	Besonders detaillierte Diagnoseinformation

Wichtige Bewertungsregeln

- Nicht jeder Fehler im Protokoll verursacht die untersuchte Störung.
- Eine hohe Zahl von Warnungen bedeutet nicht automatisch ein schwerwiegendes Problem.
- Ereignis-ID und Quelle müssen gemeinsam betrachtet werden.
- Derselbe Ereigniscode kann bei unterschiedlichen Quellen etwas anderes bedeuten.
- Der Zeitstempel muss zur tatsächlichen Störung passen.
- Folgefehler dürfen nicht mit der ursprünglichen Ursache verwechselt werden.
- Ereignisse vor dem sichtbaren Fehler sind häufig besonders wichtig.
- Ein unerwarteter Neustart kann ältere Protokolle oder flüchtige Zustände verändern.

Benutzerdefinierte Ansicht erstellen

1. **Ereignisanzeige** öffnen.
2. **Benutzerdefinierte Ansichten** auswählen.
3. **Benutzerdefinierte Ansicht erstellen** öffnen.
4. Zeitraum und Ereignisstufen festlegen.
5. Benötigte Protokolle oder Quellen auswählen.
6. Filter speichern und Ergebnisse chronologisch untersuchen.

> Das bloße Löschen eines Ereignisprotokolls behebt keine Ursache und vernichtet möglicherweise wichtige Diagnoseinformationen.

4. Windows-Ereignisse mit PowerShell untersuchen

Verfügbare klassische Protokolle anzeigen

```
[RO] Get-WinEvent -ListLog * |
    Select-Object LogName, RecordCount, IsEnabled
```

Letzte 50 Systemereignisse

```
[RO][SENS] Get-WinEvent -LogName System -MaxEvents 50
```

Fehler und kritische Ereignisse der letzten 24 Stunden

```
[RO][SENS] Get-WinEvent -FilterHashtable @{
    LogName   = 'System'
    Level     = 1, 2
    StartTime = (Get-Date).AddHours(-24)
} | Select-Object TimeCreated, Id, ProviderName, LevelDisplayName, Message
```

Ereignisse einer bestimmten Quelle

```
[RO][SENS] Get-WinEvent -FilterHashtable @{
    LogName      = 'System'
    ProviderName = 'Service Control Manager'
    StartTime    = (Get-Date).AddDays(-1)
} | Select-Object TimeCreated, Id, LevelDisplayName, Message
```

Bestimmte Ereignis-ID suchen

```
[RO][SENS] Get-WinEvent -FilterHashtable @{
    LogName   = 'System'
    Id        = 6008
    StartTime = (Get-Date).AddDays(-7)
}
```

Ergebnisse exportieren

```
[RO][SENS] Get-WinEvent -LogName System -MaxEvents 500 |
    Export-Csv -Path ".\systemereignisse.csv" -NoTypeInformation -Encoding UTF8
```

> Exportierte Ereignisse können Benutzernamen, Computerbezeichnungen, Pfade, IP-Adressen und Anwendungsdaten enthalten.

Vorteil von `FilterHashtable`

Die Filterung erfolgt bereits beim Abruf. Das ist bei großen Ereignisprotokollen meist effizienter, als zunächst alle Ereignisse einzulesen und sie anschließend mit `Where-Object` zu filtern.

5. Häufig relevante Windows-Ereignisse einordnen

Ereignis-ID	Typische Quelle	Grundsätzliche Bedeutung
`41`	Kernel-Power	Windows wurde ohne reguläres Herunterfahren neu gestartet
`1074`	User32	Ein Prozess oder Benutzer leitete Herunterfahren oder Neustart ein
`6005`	EventLog	Ereignisprotokolldienst wurde gestartet
`6006`	EventLog	Ereignisprotokolldienst wurde regulär beendet
`6008`	EventLog	Vorheriges Herunterfahren war unerwartet
`7031`	Service Control Manager	Ein Dienst wurde unerwartet beendet

Ereignis-ID	Typische Quelle	Grundsätzliche Bedeutung
7034	Service Control Manager	Ein Dienst wurde unerwartet beendet
7040	Service Control Manager	Starttyp eines Dienstes wurde geändert
7045	Service Control Manager	Ein Dienst wurde im System installiert

> Die Ereignis-ID allein ist kein vollständiger Befund. Quelle, Meldung, Zeitpunkt, betroffene Komponente und benachbarte Ereignisse müssen zusätzlich ausgewertet werden.

Ein Ereignis `Kernel-Power 41` bestätigt beispielsweise, dass Windows zuvor nicht ordnungsgemäß heruntergefahren wurde. Es beweist jedoch nicht, ob Stromverlust, Hardwarefehler, Absturz, erzwungenes Ausschalten oder eine andere Ursache verantwortlich war.

6. Zuverlässigkeitsverlauf auswerten

Zuverlässigkeitsüberwachung öffnen

```
[RO] perfmon /rel
```

Der Zuverlässigkeitsverlauf stellt wichtige Ereignisse nach Tagen geordnet dar. Dazu können gehören:

- Anwendungsabstürze
- Windows-Fehler
- fehlgeschlagene Updates
- Treiberinstallationen
- Softwareinstallationen
- erfolgreiche Updates
- Hardwarefehler

Sinnvolle Vorgehensweise

1. Tag und Uhrzeit der Störung auswählen.
2. Kritische Ereignisse untersuchen.
3. Gleichzeitig installierte Updates oder Anwendungen beachten.
4. Technische Details öffnen.

5. Befund mit Ereignisanzeige und Anwendungsprotokollen vergleichen.

> Der Stabilitätsindex ist eine verdichtete Kennzahl. Er beweist nicht, welche Komponente die Ursache einer Störung ist.

7. Task-Manager zur ersten Eingrenzung verwenden

Task-Manager öffnen

```
[RO] taskmgr
```

Wichtige Bereiche

Registerkarte	Verwendung
Prozesse	Momentane CPU-, RAM-, Datenträger-, Netzwerk- und GPU-Nutzung
Leistung	Gesamtauslastung und zeitliche Kurzansicht
App-Verlauf	Ressourcennutzung unterstützter Apps
Autostart-Apps	Programme beim Benutzerstart
Benutzer	Ressourcen nach angemeldeten Benutzern
Details	Prozesse mit PID, Status, Benutzer und weiteren Spalten
Dienste	Dienststatus und zugehörige Informationen

Bei hoher Auslastung prüfen

- Ist die Auslastung dauerhaft oder nur kurzfristig?
- Welcher Prozess verursacht sie?
- Gehört der Prozess zu einer Anwendung oder einem Windows-Dienst?
- Steigt der Speicherverbrauch fortlaufend?
- Liegt die Datenträgeraktivität bei hoher Antwortzeit?
- Wird ein Prozess mehrfach ausgeführt?
- Ist die Auslastung Folge eines Updates, Scans oder Backups?
- Stimmt der Fehlerzeitpunkt mit der Auslastung überein?

> Das sofortige Beenden eines unbekannten Prozesses kann Datenverlust verursachen oder Systemfunktionen unterbrechen.

8. Prozesse mit PowerShell untersuchen

Prozesse anzeigen

```
[RO][SENS] Get-Process
```

Prozesse nach CPU-Zeit sortieren

```
[RO][SENS] Get-Process |
    Sort-Object CPU -Descending |
    Select-Object -First 15 Name, Id, CPU, WorkingSet64
```

`CPU` enthält die bisher verbrauchte Prozessorzeit und nicht unmittelbar die aktuelle prozentuale CPU-Auslastung.

Prozesse nach Arbeitsspeicher sortieren

```
[RO][SENS] Get-Process |
    Sort-Object WorkingSet64 -Descending |
    Select-Object -First 15 Name,
        Id,
        @{Name='RAM_MiB';Expression={[math]::Round($_.WorkingSet64 / 1MB, 1)}}
```

Bestimmten Prozess anzeigen

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

Prozessdetails über CIM

```
[RO][SENS] Get-CimInstance Win32_Process -Filter "ProcessId=<PID>" |
    Select-Object ProcessId, ParentProcessId, Name, ExecutablePath, CommandLine
```

Die Befehlszeile kann Pfade, Benutzerdaten, Servernamen oder sogar ungeschützt übergebene Zugangsdaten enthalten.

Prozess kontrolliert beenden

```
[CHANGE] Stop-Process -Id <PID>
```

Erzwungenes Beenden

```
[CHANGE] Stop-Process -Id <PID> -Force
```

> `-Force` darf nicht als allgemeine Fehlerbehebung verwendet werden. Nicht gespeicherte Daten können verloren gehen und abhängige Komponenten können ausfallen.

9. Ressourcenmonitor verwenden

Ressourcenmonitor öffnen

```
[RO] resmon
```

Der Ressourcenmonitor zeigt detaillierte Informationen zu:

- CPU
- Arbeitsspeicher
- Datenträgeraktivität
- Netzwerkaktivität
- TCP-Verbindungen
- lauschenden Ports
- Prozessen und zugehörigen Dateien

Typische Anwendungsfälle

- Prozess mit hoher Datenträgeraktivität bestimmen
- Datei mit vielen Zugriffen ermitteln
- Prozess hinter einer TCP-Verbindung identifizieren
- lauschenden Port einer Anwendung feststellen
- Hard Faults und Speichernutzung untersuchen

- Abhängigkeiten eines angehaltenen Prozesses betrachten

> Ein Hard Fault ist nicht automatisch ein Hardwarefehler. Er bezeichnet in diesem Zusammenhang einen Speicherzugriff, bei dem benötigte Daten nicht im physischen Arbeitsspeicher vorlagen und aus einer anderen Speicherquelle geladen werden mussten.

10. Leistungsüberwachung und Performance Counter

Leistungsüberwachung öffnen

```
[RO] perfmon
```

Die Leistungsüberwachung kann Messwerte in Echtzeit anzeigen oder über Datenkollektorsätze längerfristig erfassen.

Verfügbare Leistungsindikatoren mit PowerShell anzeigen

```
[RO] Get-Counter -ListSet *
```

Einzelne Messung durchführen

```
[TEST] Get-Counter '\Processor(_Total)\% Processor Time'
```

Mehrere Messwerte erfassen

```
[TEST] Get-Counter `
    '\Processor(_Total)\% Processor Time',
    '\Memory\Available MBytes',
    '\PhysicalDisk(_Total)\Avg. Disk sec/Transfer' `
    -SampleInterval 2 `
    -MaxSamples 10
```

Wichtige Kategorien

Kategorie	Beispiele

Processor	Gesamtauslastung und Prozessorzeit
Memory	verfügbarer Speicher und Paging
PhysicalDisk	Warteschlangen, Übertragungen und Latenz
Network Interface	übertragene und verworfene Daten
Process	Werte einzelner Prozesse
System	Prozesse, Threads und Systemaktivität

Bewertungsregeln

- Einzelmessungen bilden nur einen kurzen Zeitpunkt ab.
- Grenzwerte hängen von Arbeitslast, Hardware und Anwendung ab.
- Durchschnittswerte können kurze Lastspitzen verdecken.
- Eine hohe Warteschlange beweist ohne Latenz- und Durchsatzwerte noch keinen Datenträgerdefekt.
- Prozessinstanzen können sich bei einem Neustart oder mehreren gleichnamigen Prozessen ändern.
- Messung und Fehler müssen zeitlich zusammenpassen.

11. Dienste untersuchen

Diensteverwaltung öffnen

```
[RO] services.msc
```

Alle Dienste mit PowerShell anzeigen

```
[RO] Get-Service
```

Beendete Dienste anzeigen

```
[RO] Get-Service |
    Where-Object Status -eq 'Stopped'
```

Ein beendeter Dienst ist nicht automatisch fehlerhaft. Viele Dienste werden nur bei Bedarf gestartet.

Bestimmten Dienst prüfen

```
[RO] Get-Service -Name "<DIENSTNAME>"
```

Ausführlichere Dienstinformationen

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

Dienstkonfiguration mit `sc.exe`

```
[RO] sc.exe qc "<DIENSTNAME>"
```

Dienstabhängigkeiten

```
[RO] Get-Service -Name "<DIENSTNAME>" -RequiredServices
```

Abhängige Dienste

```
[RO] Get-Service -Name "<DIENSTNAME>" -DependentServices
```

Dienst starten

```
[CHANGE][PRIV] Start-Service -Name "<DIENSTNAME>"
```

Dienst neu starten

```
[CHANGE][PRIV][RESTART] Restart-Service -Name "<DIENSTNAME>"
```

> Vor einem Dienstneustart müssen abhängige Anwendungen, aktive Benutzer, Transaktionen und Auswirkungen auf andere Systeme geprüft werden.

12. Netzwerkgrundlagen erfassen

Vollständige IP-Konfiguration

```
[RO][SENS] ipconfig /all
```

PowerShell-Netzwerkkonfiguration

```
[RO][SENS] Get-NetIPConfiguration
```

IP-Adressen anzeigen

```
[RO][SENS] Get-NetIPAddress
```

Netzwerkadapter anzeigen

```
[RO] Get-NetAdapter |
    Select-Object Name, InterfaceDescription, Status, LinkSpeed, MacAddress
```

Routingtabelle anzeigen

```
[RO][SENS] route print
```

Alternativ:

```
[RO][SENS] Get-NetRoute |
    Sort-Object InterfaceIndex, DestinationPrefix
```

DNS-Server anzeigen

```
[RO][SENS] Get-DnsClientServerAddress
```

Wichtige Prüfwerte

- erwartete IPv4- oder IPv6-Adresse
- Präfix beziehungsweise Subnetzmaske
- Standardgateway
- DNS-Server
- DHCP-Status
- Adapterstatus
- Verbindungsgeschwindigkeit
- unerwartete virtuelle Adapter
- VPN- oder Hypervisor-Routen
- APIPA-Adresse aus `169.254.0.0/16`

> Eine APIPA-Adresse kann darauf hinweisen, dass keine erwartete IPv4-Konfiguration per DHCP bezogen wurde. Sie beweist jedoch nicht allein, warum der DHCP-Vorgang scheiterte.

13. Erreichbarkeit und Netzwerkpfad testen

Lokalen TCP/IP-Stack testen

```
[TEST] ping 127.0.0.1
```

Standardgateway testen

```
[TEST] ping <GATEWAY-IP>
```

Ziel testen

```
[TEST] ping <ZIEL>
```

Route zum Ziel verfolgen

```
[TEST][SENS] tracert <ZIEL>
```

Verlust und Laufzeit entlang des Pfades untersuchen

```
[TEST][SENS] pathping <ZIEL>
```

TCP-Port mit PowerShell testen

```
[TEST][SENS] Test-NetConnection -ComputerName "<ZIEL>" -Port <PORT>
```

Beispiel:

```
[TEST][SENS] Test-NetConnection -ComputerName "server.example" -Port 443
```

Erweiterte Verbindungsinformationen

```
[TEST][SENS] Test-NetConnection -ComputerName "<ZIEL>" -DiagnoseRouting -InformationLevel Detailed
```

Bewertungsregeln

- Ein fehlgeschlagener Ping beweist nicht, dass das Ziel ausgefallen ist.
- ICMP kann gefiltert werden, während ein Anwendungsport erreichbar bleibt.
- Ein erfolgreicher Ping beweist nicht, dass die Anwendung funktioniert.
- Ein erfolgreicher TCP-Test bestätigt die Verbindung zum getesteten Port, aber nicht automatisch eine korrekte Anwendungssitzung.
- Zeitüberschreitungen bei `tracert` können durch gefilterte Antworten entstehen.
- Paketverlust an einem Zwischenknoten ist nur dann aussagekräftig, wenn er sich bis zum Ziel fortsetzt.

14. DNS-Auflösung untersuchen

DNS-Abfrage mit PowerShell

```
[TEST][SENS] Resolve-DnsName "<HOSTNAME>"
```

Bestimmten DNS-Server abfragen

```
[TEST][SENS] Resolve-DnsName "<HOSTNAME>" -Server "<DNS-SERVER>"
```

Klassische DNS-Abfrage

```
[TEST][SENS] nslookup <HOSTNAME>
```

DNS-Clientcache anzeigen

```
[RO][SENS] Get-DnsClientCache
```

Alternativ:

```
[RO][SENS] ipconfig /displaydns
```

DNS-Clientcache leeren

```
[CHANGE][PRIV] Clear-DnsClientCache
```

Alternativ:

```
[CHANGE][PRIV] ipconfig /flushdns
```

Typische Fehlerquellen

- falscher DNS-Server
- falscher oder abgelaufener DNS-Eintrag
- Suchsuffix fehlt
- Split-DNS liefert abhängig vom Standort andere Antworten
- VPN überschreibt DNS-Einstellungen
- lokaler Cache enthält einen alten Eintrag
- Hosts-Datei überschreibt DNS
- Firewall blockiert DNS-Abfragen
- IPv4- und IPv6-Antworten führen zu unterschiedlichen Pfaden

> Das Leeren des DNS-Caches sollte nur erfolgen, wenn ein veralteter lokaler Cache als Ursache infrage kommt. Es behebt keine falschen Einträge auf dem DNS-Server.

15. Ports und Verbindungen untersuchen

TCP-Verbindungen anzeigen

```
[RO][SENS] Get-NetTCPConnection
```

Lauschende TCP-Ports

```
[RO][SENS] Get-NetTCPConnection -State Listen |
    Sort-Object LocalPort |
    Select-Object LocalAddress, LocalPort, OwningProcess
```

Zugehörigen Prozess bestimmen

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

Klassische Portübersicht

```
[RO][SENS] netstat -ano
```

Lauschende Ports

```
[RO][SENS] netstat -ano | findstr LISTENING
```

Ausführbare Programme einbeziehen

```
[RO][PRIV][SENS] netstat -abno
```

UDP-Endpunkte

```
[RO][SENS] Get-NetUDPEndpoint |
    Sort-Object LocalPort
```

Wichtige Zustände

Zustand	Bedeutung

`Listen`	Prozess wartet auf eingehende TCP-Verbindungen
`Established`	TCP-Verbindung besteht
`TimeWait`	Verbindung wurde beendet und bleibt vorübergehend gespeichert
`SynSent`	Verbindungsaufbau wurde begonnen, Antwort steht noch aus
`CloseWait`	Gegenstelle hat beendet; lokaler Prozess muss noch schließen

> Ein offener Hostport beweist noch nicht, dass die Anwendung fachlich korrekt arbeitet. Zusätzlich müssen Protokoll, Antwort und gegebenenfalls Authentifizierung getestet werden.

16. Windows-Firewall untersuchen

Firewallprofile anzeigen

```
[RO] Get-NetFirewallProfile |
    Select-Object Name, Enabled, DefaultInboundAction, DefaultOutboundAction
```

Aktivierte Firewallregeln anzeigen

```
[RO][SENS] Get-NetFirewallRule -Enabled True |
    Select-Object DisplayName, Direction, Action, Profile
```

Regeln anhand eines Namens suchen

```
[RO][SENS] Get-NetFirewallRule |
    Where-Object DisplayName -Like '*<SUCHBEGRIFF>*'
```

Firewallverwaltung öffnen

```
[RO] wf.msc
```

Typische Prüfpunkte

- aktives Netzwerkprofil
- Eingangs- oder Ausgangsrichtung
- erlaubende und blockierende Regeln
- lokaler Port und Remoteport
- lokales und entferntes Adressnetz
- zugeordnetes Programm oder Dienst
- Domänen-, privates oder öffentliches Profil
- zentral durch Gruppenrichtlinie verwaltete Regeln

> Die Firewall darf nicht pauschal deaktiviert werden, um ein Verbindungsproblem zu testen. Stattdessen muss die konkrete Verbindung mit passender Protokollierung und eng begrenzten Regeln untersucht werden.

17. Datenträger, Volumes und Speicherplatz prüfen

Volumes anzeigen

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

Datenträger anzeigen

```
[RO] Get-Disk
```

Partitionen anzeigen

```
[RO] Get-Partition
```

Freien Speicher übersichtlich berechnen

```
[RO] Get-Volume |
    Where-Object DriveLetter |
    Select-Object DriveLetter,
        FileSystemLabel,
        @{Name='Frei_GiB';Expression={[math]::Round($_.SizeRemaining / 1GB, 2)}},
        @{Name='Gesamt_GiB';Expression={[math]::Round($_.Size / 1GB, 2)}}
```

Datenträgerverwaltung öffnen

```
[RO] diskmgmt.msc
```

Dateisystem online prüfen

```
[TEST][PRIV] chkdsk C: /scan
```

Wichtige Unterscheidungen

- Freier Speicherplatz ist nicht dasselbe wie Datenträgerzustand.
- Ein Zustand `Healthy` schließt nicht jede Hardwarestörung aus.
- Ein voller Datenträger kann Updates, Protokollierung, Datenbanken und Benutzeranmeldungen beeinträchtigen.
- Eine hohe Datenträgerauslastung bedeutet nicht automatisch, dass das Laufwerk defekt ist.
- Dateisystemfehler, Hardwarefehler, Controllerprobleme und Anwendungs-I/O müssen getrennt untersucht werden.

> Reparaturoptionen wie `chkdsk /f` oder `/r` können lange Laufzeiten, exklusive Datenträgerzugriffe oder einen Neustart erfordern. Sie dürfen nicht ohne Sicherung und Wartungsplanung eingesetzt werden.

18. Arbeitsspeicher untersuchen

Grundlegende Speicherinformationen

```
[RO] Get-CimInstance Win32_OperatingSystem |
    Select-Object @{
```

```
    Name='Gesamt_GiB'
    Expression={[math]::Round($_.TotalVisibleMemorySize / 1MB, 2)}
  }, @{
    Name='Frei_GiB'
    Expression={[math]::Round($_.FreePhysicalMemory / 1MB, 2)}
  }
```

Installierte Speichermodule

```
[RO][SENS] Get-CimInstance Win32_PhysicalMemory |
  Select-Object BankLabel,
                DeviceLocator,
                Capacity,
                Speed,
                Manufacturer,
                PartNumber
```

Windows-Speicherdiagnose öffnen

```
[TEST][RESTART] mdsched.exe
```

Die Speicherdiagnose bietet normalerweise einen Test beim nächsten Neustart oder einen sofortigen Neustart mit anschließender Prüfung an.

> Ein Speicherdiagnosetest unter Last oder ein einzelner erfolgreicher Durchlauf schließt sporadische Hardwarefehler nicht zwingend aus.

19. Geräte und Treiber prüfen

Geräte-Manager öffnen

```
[RO] devmgmt.msc
```

Vorhandene Plug-and-Play-Geräte

```
[RO][SENS] Get-PnpDevice
```

Geräte mit Fehlerstatus

```
[RO][SENS] Get-PnpDevice |
    Where-Object Status -ne 'OK'
```

Installierte Treiber

```
[RO][SENS] driverquery /v
```

Signierte Treiber mit PowerShell

```
[RO][SENS] Get-CimInstance Win32_PnPSignedDriver |
    Select-Object DeviceName,
                  DriverVersion,
                  DriverDate,
                  Manufacturer,
                  InfName
```

Typische Prüfpunkte

- Gerätestatus und Fehlercode
- Treiberversion und Treiberdatum
- kürzlich erfolgte Treiberänderung
- unbekannte oder deaktivierte Geräte
- Hardware-ID
- Herstellerfreigabe für die Windows-Version
- Zusammenhang zwischen Treiberinstallation und Fehlerzeitpunkt

> Ein neuerer Treiber ist nicht automatisch geeigneter. Entscheidend sind Hardwaremodell, Betriebssystemversion und Freigabe durch Hersteller oder Geräteanbieter.

20. Autostart, Aufgaben und Systemstart untersuchen

Autostartprogramme über CIM anzeigen

```
[RO][SENS] Get-CimInstance Win32_StartupCommand |
    Select-Object Name, Command, Location, User
```

Aufgabenplanung öffnen

```
[RO] taskschd.msc
```

Geplante Aufgaben anzeigen

```
[RO][SENS] Get-ScheduledTask |
    Select-Object TaskPath, TaskName, State
```

Aufgaben mit letztem Ergebnis

```
[RO][SENS] Get-ScheduledTask | ForEach-Object {
    $info = $_ | Get-ScheduledTaskInfo
    [PSCustomObject]@{
        TaskPath       = $_.TaskPath
        TaskName       = $_.TaskName
        State          = $_.State
        LastRunTime    = $info.LastRunTime
        LastTaskResult = $info.LastTaskResult
        NextRunTime    = $info.NextRunTime
    }
}
```

Systemkonfiguration öffnen

```
[RO] msconfig
```

> Autostarteinträge und geplante Aufgaben dürfen nicht wahllos deaktiviert werden. Sie können zu Sicherheitssoftware, Backups, Updates, Treibern oder geschäftskritischen Anwendungen gehören.

21. Windows-Updates prüfen

Installierte Hotfixes anzeigen

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

Updateverlauf in den Einstellungen

```
Einstellungen
→ Windows Update
→ Updateverlauf
```

Windows-Update-Ereignisse abrufen

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

Typische Prüfpunkte

- Zeitpunkt der Installation
- KB-Nummer
- erfolgreicher oder fehlgeschlagener Abschluss
- erforderlicher Neustart
- Treiberupdate
- bekannte Abhängigkeit zur betroffenen Anwendung
- wiederholt fehlgeschlagene Installation
- ausreichend freier Speicher
- Richtlinien oder Updateverwaltung im Unternehmen

> Ein zeitlicher Zusammenhang zwischen Update und Fehler ist ein wichtiger Hinweis, aber noch kein Ursachennachweis. Vor einer Deinstallation müssen bekannte Probleme, Sicherheitsfolgen und Herstellerangaben geprüft werden.

22. Windows-Systemdateien mit SFC prüfen

Der System File Checker überprüft geschützte Windows-Systemdateien und kann beschädigte Dateien ersetzen.

Systemdateien prüfen und reparieren

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

Nur eine bestimmte Datei prüfen

```
[TEST][PRIV] sfc /verifyfile="<VOLLSTÄNDIGER_PFAD>"
```

Nur prüfen, ohne Reparatur

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

Mögliche Ergebnisse

- Keine Integritätsverletzungen gefunden
- Beschädigte Dateien gefunden und erfolgreich repariert
- Beschädigte Dateien gefunden, aber nicht alle repariert
- Prüfung konnte nicht ausgeführt werden

SFC-Einträge aus dem CBS-Protokoll filtern

```
[RO][PRIV][SENS] findstr /c:"[SR]" %windir%\Logs\CBS\CBS.log
```

> SFC ist kein universelles Reparaturprogramm. Es untersucht geschützte Windows-Systemdateien, nicht automatisch Anwendungsdateien, Benutzerdaten, Hardware oder sämtliche Konfigurationen.

23. Windows-Komponentenspeicher mit DISM prüfen

DISM kann den Zustand des Windows-Abbilds beziehungsweise Komponentenspeichers untersuchen und reparieren.

Schnelle Statusprüfung

```
[RO][PRIV] DISM /Online /Cleanup-Image /CheckHealth
```

Ausführliche Prüfung

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

Windows-Abbild reparieren

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

Bedeutung der Parameter

Parameter	Bedeutung
`/Online`	Aktuell laufendes Windows wird bearbeitet
`/Cleanup-Image`	Wartungsfunktion für das Windows-Abbild
`/CheckHealth`	Prüft, ob eine Beschädigung bereits erkannt wurde
`/ScanHealth`	Führt eine ausführlichere Prüfung durch
`/RestoreHealth`	Versucht erkannte Beschädigungen zu reparieren

Wichtige Hinweise

- `/RestoreHealth` verändert den Systemzustand.
- Der Vorgang kann längere Zeit dauern.
- Eine Reparaturquelle kann erforderlich sein.
- Die Quelle muss zum installierten Windows passen.
- Fehler müssen anhand des DISM-Protokolls ausgewertet werden.
- Nach erfolgreicher Reparatur kann eine erneute SFC-Prüfung sinnvoll sein.

DISM-Protokoll

```
C:\Windows\Logs\DISM\dism.log
```

CBS-Protokoll

```
C:\Windows\Logs\CBS\CBS.log
```

> DISM und SFC dürfen nicht reflexartig bei jedem Windows-Problem ausgeführt werden. Zuerst sollte ein Zusammenhang mit beschädigten Windows-Komponenten oder Systemdateien begründet werden.

24. Gruppenrichtlinien und Richtlinienanwendung prüfen

Angewendete Richtlinien anzeigen

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

Ausführlichen HTML-Bericht erzeugen

```
[RO][SENS] gpresult /h ".\gpresult.html"
```

Resultant Set of Policy öffnen

```
[RO][SENS] rsop.msc
```

Gruppenrichtlinien-Ereignisse

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

Gruppenrichtlinien aktualisieren

```
[CHANGE][PRIV] gpupdate /force
```

> `gpupdate /force` kann Richtlinien erneut anwenden, Anmelde- oder Neustartanforderungen auslösen und den Arbeitszustand beeinflussen. Der Befehl ist kein Ersatz für die Analyse einer fehlerhaften Richtlinie.

25. Anmeldungen und Benutzerkontext untersuchen

Aktuellen Benutzer anzeigen

```
[RO] whoami
```

Gruppen und Sicherheitskennungen anzeigen

```
[RO][SENS] whoami /groups
```

Aktuelle Berechtigungen anzeigen

```
[RO][SENS] whoami /priv
```

Angemeldete Sitzungen

```
[RO][SENS] query user
```

Lokale Benutzer anzeigen

```
[RO][SENS] Get-LocalUser
```

Lokale Gruppenmitgliedschaften

```
[RO][SENS] Get-LocalGroupMember -Group "<GRUPPENNAME>"
```

Kerberos-Tickets anzeigen

```
[RO][SENS] klist
```

Typische Ursachen kontextabhängiger Fehler

- anderer Benutzer
- fehlende Gruppenmitgliedschaft
- nicht aktualisiertes Zugriffstoken
- lokale statt Domänenanmeldung
- abgelaufenes oder fehlendes Kerberos-Ticket

- Dienst läuft unter anderem Konto
- unterschiedliche Umgebungsvariablen
- abweichendes Benutzerprofil
- erhöhte und nicht erhöhte Sitzung besitzen unterschiedliche Rechte

> Ein erfolgreicher Test als Administrator beweist nicht, dass die Funktion für den vorgesehenen Benutzer korrekt berechtigt ist.

26. Remotedienste und Freigaben prüfen

SMB-Freigaben anzeigen

```
[RO][SENS] Get-SmbShare
```

Bestehende SMB-Verbindungen

```
[RO][SENS] Get-SmbConnection
```

Erreichbarkeit von SMB testen

```
[TEST][SENS] Test-NetConnection -ComputerName "<SERVER>" -Port 445
```

Remotedesktopport testen

```
[TEST][SENS] Test-NetConnection -ComputerName "<SERVER>" -Port 3389
```

WinRM-Konfiguration anzeigen

```
[RO][SENS] winrm get winrm/config
```

WinRM-Verbindung testen

```
[TEST][SENS] Test-WSMan -ComputerName "<SERVER>"
```

Wichtige Unterscheidung

Eine erfolgreiche TCP-Verbindung bestätigt nur, dass der getestete Port erreichbar ist. Sie bestätigt nicht automatisch:

- erfolgreiche Authentifizierung
- korrekte Freigabeberechtigung
- NTFS-Berechtigung
- funktionierende Remotedesktopsitzung
- gültiges Zertifikat
- erfolgreiche Anwendungskommunikation

27. Abstürze und Speicherabbilder untersuchen

Windows kann bei System- oder Anwendungsabstürzen Speicherabbilder erzeugen.

Typische Speicherorte

```
C:\Windows\MEMORY.DMP
C:\Windows\Minidump\
%LOCALAPPDATA%\CrashDumps\
```

Die tatsächliche Konfiguration und der Speicherort können abweichen.

Start- und Wiederherstellungseinstellungen öffnen

```
[RO] sysdm.cpl
```

Danach:

```
Erweitert
→ Starten und Wiederherstellen
→ Einstellungen
```

Typische Abbildarten

Typ	Inhalt
Kleines Speicherabbild	Begrenzte Informationen zum Absturz
Kernelspeicherabbild	Kernelbezogene Speicherbereiche
Automatisches Speicherabbild	Von Windows verwaltete Auswahl

Typ	Inhalt
Vollständiges Speicherabbild	Umfassender Inhalt des physischen Speichers
Aktives Speicherabbild	Für aktive Kernel- und Systembereiche optimiert

Wichtige Sicherheitsregel

Speicherabbilder können enthalten:

- Benutzerdaten
- Dateiinhalte
- Prozessspeicher
- Schlüsselmaterial
- Tokens
- Passwörter oder andere Geheimnisse
- Netzwerk- und Sitzungsinformationen

> Speicherabbilder müssen wie sensible Diagnosedaten behandelt und dürfen nicht ungeprüft weitergegeben werden.

Zur detaillierten Analyse kann Microsoft WinDbg eingesetzt werden. WinDbg wird auf einer späteren Seite zusammen mit weiteren spezialisierten Werkzeugen behandelt.

28. Abgesicherter Modus und sauberer Systemstart

Der abgesicherte Modus startet Windows mit einer eingeschränkten Auswahl von Treibern und Diensten. Damit kann geprüft werden, ob ein Fehler durch zusätzliche Software, Treiber oder Autostartkomponenten beeinflusst wird.

Ein sauberer Systemstart deaktiviert gezielt nicht benötigte Dienste und Autostartprogramme, um Konflikte einzugrenzen.

Systemkonfiguration öffnen

```
[CHANGE][RESTART] msconfig
```

Wichtige Risiken

- Sicherheits- oder Verwaltungssoftware kann vorübergehend deaktiviert werden.
- Geschäftsanwendungen oder Hintergrunddienste können ausfallen.
- Änderungen müssen dokumentiert und anschließend zurückgenommen werden.

- Abgesicherter Modus und sauberer Start sind Diagnosezustände, keine dauerhafte Lösung.
- In verwalteten Umgebungen kann eine Freigabe erforderlich sein.

Geeignete Fragestellung

```
Tritt der Fehler auch mit einer minimalen Auswahl von Treibern,
Diensten und Autostartprogrammen auf?
```

Ein ausbleibender Fehler grenzt die Ursache ein, identifiziert aber noch nicht automatisch die verantwortliche Komponente.

29. Fernzugriff und Datensammlung sicher durchführen

Vor einer Ferndiagnose müssen geklärt werden:

- korrekter Zielcomputer
- berechtigte Person
- Wartungs- oder Diagnosefenster
- Auswirkungen auf aktive Benutzer
- erlaubte Werkzeuge
- Speicherort der Diagnosedaten
- Schutz sensibler Informationen
- Rückfallplan für Änderungen

Computername prüfen

```
[RO] $env:COMPUTERNAME
```

Benutzerkontext prüfen

```
[RO] whoami
```

PowerShell-Remotesitzung testen

```
[TEST][SENS] Test-WSMan -ComputerName "<COMPUTER>"
```

Befehl remote ausführen

```
[RO][SENS] Invoke-Command -ComputerName "<COMPUTER>" -ScriptBlock {
    Get-CimInstance Win32_OperatingSystem |
        Select-Object Caption, Version, LastBootUpTime
}
```

> Remoteausgaben, Transkripte und Diagnosearchive können sensible Daten enthalten. Sie müssen zweckgebunden gespeichert, geschützt und nach den geltenden Vorgaben gelöscht werden.

30. Gefährliche Schnelllösungen vermeiden

Maßnahme	Risiko
Unbekannten Prozess sofort beenden	Datenverlust oder Dienstausfall
Dienst ohne Prüfung neu starten	Unterbrechung abhängiger Anwendungen
Windows-Firewall vollständig deaktivieren	Schutzwirkung entfällt
Ereignisprotokolle löschen	Diagnosenachweise gehen verloren
Registrywerte auf Verdacht ändern	System oder Anwendung wird beschädigt
Treiber ungeprüft aktualisieren	Inkompatibilität oder Startfehler
Update sofort deinstallieren	Sicherheitslücke oder Folgefehler
`chkdsk /f` ungeplant ausführen	Neustart und lange Ausfallzeit
DISM oder SFC als Universallösung verwenden	Ursache bleibt ungeklärt
Autostart und Dienste wahllos deaktivieren	Sicherheits- und Geschäftsfunktionen fallen aus
Speicherabbild unredigiert weitergeben	Geheimnisse und Benutzerdaten werden offengelegt
Computer hart ausschalten	Daten- und Dateisystemschäden
Systemwiederherstellung ungeprüft starten	Neuere Änderungen können verloren gehen
„PC zurücksetzen“ verwenden	Anwendungen, Einstellungen oder Daten gehen verloren

Vor einer Änderung prüfen

1. Ist die Ursache ausreichend eingegrenzt?
2. Welche Systeme und Benutzer sind betroffen?
3. Existiert eine aktuelle Sicherung?
4. Ist die Maßnahme reversibel?
5. Welche Abhängigkeiten bestehen?

6. Ist ein Neustart erforderlich?
7. Gibt es ein Wartungsfenster?
8. Wie wird die Funktion anschließend getestet?
9. Wie wird die Änderung dokumentiert?

31. Sichere Reihenfolge einer Windows-Fehleranalyse

1. Betroffenen Computer und Benutzer eindeutig bestimmen.
2. Originale Fehlermeldung und Fehlerzeitpunkt erfassen.
3. Umfang und Reproduzierbarkeit prüfen.
4. Windows-Version, Build und letzte Startzeit erfassen.
5. Änderungen, Updates und Installationen vor dem Fehler feststellen.
6. Ereignisanzeige und Zuverlässigkeitsverlauf zum Fehlerzeitpunkt prüfen.
7. Prozesse, Dienste und aktuelle Ressourcenbelastung untersuchen.
8. Freien Speicherplatz und Datenträgerzustand kontrollieren.
9. Netzwerk, DNS, Route und benötigte Ports getrennt testen.
10. Benutzerkontext, Berechtigungen und Richtlinien berücksichtigen.
11. Treiber, Geräte und Abhängigkeiten prüfen.
12. Erkenntnisse aus mehreren Datenquellen abgleichen.
13. Sicherung und Rückfallplan kontrollieren.
14. Kleinste geeignete Änderung durchführen.
15. Interne und externe Funktionsprüfung ausführen.
16. Ereignisse und Ressourcen nach der Maßnahme erneut prüfen.
17. Ursache, Maßnahme und Ergebnis dokumentieren.

32. Schnelle Befehlsübersicht

Aufgabe	Befehl
Windows-Version	[RO] winver
Systeminformationen	[RO][SENS] systeminfo
Ereignisanzeige	[RO] eventvwr.msc
Systemereignisse	[RO][SENS] Get-WinEvent -LogName System -MaxEvents 50
Zuverlässigkeitsverlauf	[RO] perfmon /rel
Task-Manager	[RO] taskmgr
Ressourcenmonitor	[RO] resmon
Leistungsüberwachung	[RO] perfmon
Prozesse	[RO][SENS] Get-Process

Aufgabe	Befehl
Dienste	[RO] Get-Service
Dienstkonfiguration	[RO] sc.exe qc "<DIENSTNAME>"
IP-Konfiguration	[RO][SENS] ipconfig /all
Adapter	[RO] Get-NetAdapter
Routingtabelle	[RO][SENS] route print
Erreichbarkeit	[TEST] ping <ZIEL>
Route	[TEST][SENS] tracert <ZIEL>
TCP-Port	[TEST][SENS] Test-NetConnection "<ZIEL>" -Port <PORT>
DNS	[TEST][SENS] Resolve-DnsName "<HOSTNAME>"
TCP-Verbindungen	[RO][SENS] Get-NetTCPConnection
Ports und PID	[RO][SENS] netstat -ano
Firewallprofile	[RO] Get-NetFirewallProfile
Volumes	[RO] Get-Volume
Datenträger	[RO] Get-Disk
Geräte	[RO][SENS] Get-PnpDevice
Treiber	[RO][SENS] driverquery /v
Geplante Aufgaben	[RO][SENS] Get-ScheduledTask
Gruppenrichtlinien	[RO][SENS] gpresult /r
Benutzerkontext	[RO] whoami
Systemdateien	[CHANGE][PRIV] sfc /scannow
Abbildprüfung	[TEST][PRIV] DISM /Online /Cleanup-Image /ScanHealth
Abbildreparatur	[CHANGE][PRIV] DISM /Online /Cleanup-Image /RestoreHealth

33. Dokumentationsvorlage

Ticket:

Datum und Uhrzeit:

Bearbeiter:

Computername:

Windows-Edition:

Windows-Version:

Build:

Architektur:

Letzter Systemstart:

Benutzerkontext:

Domäne oder Arbeitsgruppe:

Originale Fehlermeldung:

Fehlerzeitpunkt:

Erstmals aufgetreten:

Reproduzierbar:

Betroffene Benutzer:

Betroffene Anwendung oder Funktion:

Umfang der Störung:

Letzte funktionierende Nutzung:

Letzte Änderung:

Installierte Updates:

Treiberänderung:

Softwareinstallation:

Richtlinienänderung:

Relevante Ereignisquellen:

Ereignis-IDs:

Zuverlässigkeitsverlauf:

Betroffene Prozesse:

Betroffene Dienste:

CPU:

Arbeitsspeicher:

Datenträger:

Freier Speicher:

IP-Konfiguration:

DNS:

Standardgateway:

Getestetes Ziel:

Getesteter Port:

Testergebnis:

```
Firewallprofil:
Firewallregel:

Gerätestatus:
Treiberversion:
Systemdateiprüfung:
DISM-Status:
Absturzabbild:
Sensible Daten redigiert:

Vermutete Ursache:
Durchgeführte Tests:
Sicherung geprüft:
Geplante Änderung:
Erwartete Auswirkung:
Rückfallplan:
Neustart erforderlich:

Ergebnis nach der Maßnahme:
Interne Funktionsprüfung:
Externe Funktionsprüfung:
Monitoring:
Weiterführende Maßnahmen:
```

Merksatz

> Windows stellt viele Diagnosewerkzeuge bereit, doch kein einzelnes Werkzeug liefert automatisch die Ursache. Eine belastbare Fehleranalyse verbindet Zeitpunkt, Ereignisse, Prozesse, Dienste, Ressourcen, Netzwerk, Änderungen und Benutzerkontext zu einem nachvollziehbaren Gesamtbild.

Quellen und weiterführende Dokumentation

- Microsoft Learn – Windows-Problembehandlungsdokumentation
- Microsoft Learn – Get-WinEvent
- Microsoft Learn – Get-Process
- Microsoft Learn – Get-Service

- Microsoft Learn – Get-Counter
- Microsoft Learn – Get-NetIPConfiguration
- Microsoft Learn – Test-NetConnection
- Microsoft Learn – Resolve-DnsName
- Microsoft Learn – Get-NetTCPConnection
- Microsoft Learn – Get-NetFirewallProfile
- Microsoft Learn – Get-Volume
- Microsoft Learn – Get-PnpDevice
- Microsoft Learn – System File Checker
- Microsoft Learn – Windows-Abbild mit DISM reparieren
- Microsoft Learn – Chkdsk
- Microsoft Learn – GPResult
- Microsoft Learn – Windows-Debuggingwerkzeuge
