1.5 Arbeitshypothesen bilden und Prüfungen priorisieren

Nachdem Störung, Umfang, Zeitpunkt, Änderungen und Ausgangszustand dokumentiert wurden, beginnt die gezielte Ursachensuche.

Dabei gilt:

> Nicht die erste plausible Erklärung beheben, sondern mehrere mögliche Ursachen aufstellen und mit geeigneten Tests unterscheiden.

Eine Arbeitshypothese ist eine vorläufige, überprüfbare Erklärung für das beobachtete Fehlerbild. Sie ist noch keine bestätigte Ursache.

Ziel dieser Seite

Nach diesem Arbeitsschritt sollten:

- gesicherte Fakten von Vermutungen getrennt sein,
- mehrere technisch plausible Ursachen vorliegen,
- widersprechende Beobachtungen berücksichtigt sein,
- für jede wichtige Hypothese ein geeigneter Test definiert sein,
- erwartete Testergebnisse vorher festgelegt sein,
- ungefährliche und aussagekräftige Tests priorisiert sein,
- Testergebnisse nachvollziehbar dokumentiert werden.

---

1. Kennzeichnung der Befehle

Kennzeichnung	Bedeutung
`[RO]`	Read-only: liest Informationen aus
`[TEST]`	Führt eine aktive Prüfung aus, beispielsweise eine DNS- oder TCP-Anfrage
`[FILE]`	Erstellt oder überschreibt eine Datei
`[PRIV]`	Benötigt möglicherweise Administrator- oder Root-Rechte
`[CHANGE]`	Verändert eine Konfiguration oder einen Betriebszustand
`[SENSITIV]`	Ausgabe kann vertrauliche Daten enthalten

Ein `[TEST]` verändert normalerweise keine Konfiguration, erzeugt aber Netzwerkverkehr und möglicherweise Protokolleinträge. Wiederholte Anmeldeversuche können beispielsweise ein Benutzerkonto sperren.

---

2. Fakten, Vermutungen und Ursachen unterscheiden

Begriff	Bedeutung	Beispiel
Symptom	Sichtbare Auswirkung der Störung	Webseite ist nicht erreichbar
Fakt	Durch Messung oder Beleg bestätigte Information	TCP-Port 443 ist von Client A nicht erreichbar
Vermutung	Nicht ausreichend begründete Annahme	„Bestimmt ist die Firewall schuld“
Hypothese	Überprüfbare mögliche Erklärung	Eine Firewall-Regel blockiert VLAN 30 zu TCP 443
Vorhersage	Erwartetes Ergebnis, wenn die Hypothese stimmt	Clients aus VLAN 40 funktionieren, VLAN 30 nicht
Test	Gezielte Prüfung der Vorhersage	Porttest aus beiden VLANs
Ursache	Durch Belege und Gegenproben bestätigte Erklärung	ACL blockiert VLAN 30 zu TCP 443
Maßnahme	Veränderung zur Beseitigung der Ursache	Korrektur der freigegebenen ACL-Regel

Wichtig

„Nach dem Neustart funktionierte es wieder“ ist zunächst nur ein Ergebnis. Daraus folgt noch nicht, warum der Fehler auftrat.

Mögliche Erklärungen wären beispielsweise:

- blockierter Prozess wurde beendet,
- Speicherleck wurde vorübergehend beseitigt,
- Netzwerkverbindung wurde neu aufgebaut,
- temporärer Cache wurde geleert,
- Konfiguration wurde beim Start neu geladen,
- abhängiger Dienst wurde in anderer Reihenfolge gestartet.

---

3. Eine präzise Problembeschreibung erstellen

Eine brauchbare Hypothese benötigt eine präzise Problembeschreibung.

Ungeeignet

`Das Netzwerk funktioniert nicht.`

Besser

`Seit ungefähr 14:35 Uhr können Windows-Clients aus VLAN 30 den Server server.example über TCP 443 nicht erreichen. DNS liefert die erwartete IP-Adresse. Clients aus VLAN 40 erreichen denselben Port weiterhin.`

Eine gute Problembeschreibung enthält:

- betroffene Funktion,
- betroffene Benutzer oder Geräte,
- nicht betroffene Benutzer oder Geräte,
- Quellnetz oder Standort,
- Zielsystem,
- Protokoll und Port,
- Beginn der Störung,
- letzte bekannte Funktion,
- Fehlermeldung,
- bereits bestätigte Fakten,
- letzte relevante Änderungen.

---

4. Aufbau einer überprüfbaren Hypothese

Eine gute Arbeitshypothese kann nach diesem Muster formuliert werden:

> Wenn `<vermutete Ursache>` vorliegt, dann müsste `<beobachtbares Ergebnis>` auftreten, weil `<technischer Zusammenhang>`. Das prüfen wir mit `<Test>`. Widerlegt wäre die Hypothese durch `<Gegenergebnis>`.

Beispiel

> Wenn eine Firewall-Regel den Verkehr aus VLAN 30 blockiert, dann müsste der TCP-Port 443 aus VLAN 30 nicht erreichbar sein, während derselbe Port aus einem freigegebenen VLAN erreichbar ist. Widerlegt wäre die Hypothese, wenn die Verbindung aus beiden VLANs identisch fehlschlägt oder die Pakete nachweislich bis zum Server gelangen.

---

5. Schlechte und gute Hypothesen

Schlechte Formulierung	Problem	Bessere Formulierung
Die Firewall ist kaputt	Zu allgemein und nicht überprüfbar	Eine Firewall-Regel blockiert TCP 443 aus VLAN 30

Schlechte Formulierung	Problem	Bessere Formulierung
Windows macht Probleme	Kein konkreter Mechanismus	Der Windows-DNS-Client verwendet einen veralteten Cacheeintrag
Der Server ist down	„Down“ ist nicht definiert	Der Webdienst lauscht nicht auf TCP 443
Es liegt am Benutzer	Keine technische Erklärung	Dem Benutzer fehlt die erforderliche Anwendungsrolle
Das Update war schuld	Nur zeitliche Korrelation	Update KB… hat den verwendeten Treiber ersetzt
DNS geht nicht	Nicht eingegrenzt	Der Client fragt einen nicht erreichbaren DNS-Server ab
Das Zertifikat ist falsch	Zu ungenau	Der aufgerufene Hostname fehlt im Subject Alternative Name
Das Netzwerk ist langsam	Keine Messgröße	Zwischen Client und Server treten Paketverluste auf

---

6. Mögliche Fehlerbereiche systematisch durchgehen

Für eine erste Hypothesenliste können folgende Bereiche verwendet werden:

Fehlerbereich	Typische Ursachen
Benutzer	Bedienfehler, falsche Eingabe, fehlende Berechtigung
Benutzerkonto	Sperrung, Ablauf, Gruppenmitgliedschaft, MFA
Client-Hardware	Netzwerkkarte, Datenträger, Arbeitsspeicher
Client-Betriebssystem	Update, Treiber, lokaler Dienst, Richtlinie
Client-Konfiguration	IP, DNS, Proxy, Zertifikat, Firewall
Anwendung	Absturz, Cache, Version, Konfiguration
Namensauflösung	DNS-Server, Eintrag, Suchdomäne, Cache
Netzwerkzugang	WLAN, Switchport, VLAN, NAC, Port-Security
Netzwerkpfad	Routing, ACL, Firewall, NAT, MTU
Zielserver	Dienst, Ressourcen, Betriebssystem, Listener
Authentifizierung	Kerberos, LDAP, RADIUS, Zertifikat, Token
Autorisierung	Rolle, Gruppe, ACL, Dateiberechtigung
Abhängiger Dienst	Datenbank, DNS, Storage, API, Queue
Sicherheitskomponente	EDR, Antivirus, Proxy, Webfilter
Automatisierung	Skript, GPO, Deployment, geplanter Job
Virtualisierung	Ressourcen, virtuelles Netzwerk, Storage

Fehlerbereich	Typische Ursachen
Cloud oder Provider	Plattformstörung, Richtlinie, Service Health
Letzte Änderung	Update, Migration, Konfiguration, Zertifikat

Nicht jeder Bereich muss ausführlich geprüft werden. Die Eingrenzung aus den vorherigen Seiten bestimmt, welche Bereiche plausibel sind.

---

7. Hypothesen aus dem Fehlerumfang ableiten

Beobachtung	Zu priorisierende Hypothesen
Nur ein Benutzer betroffen	Konto, Berechtigung, Profil, Rolle
Alle Benutzer eines Clients betroffen	Client, Betriebssystem, lokale Konfiguration
Ein Benutzer auf allen Clients betroffen	Identität, Konto, Berechtigung
Nur ein Standort betroffen	WAN, Firewall, DNS, Provider, Standortnetz
Nur ein VLAN betroffen	Routing, ACL, DHCP, VLAN-Zuordnung
Alle Clients betroffen	Server, zentraler Dienst, gemeinsame Abhängigkeit
Nur ein Server betroffen	Dienst, Betriebssystem, Ressourcen, Serverkonfiguration
Nur ein Browser betroffen	Browserprofil, Proxy, Zertifikat, Erweiterung
Zugriff per IP funktioniert, per Name nicht	DNS, Hosts-Datei, Suchdomäne
Port erreichbar, Anwendung funktioniert nicht	Protokoll, Authentifizierung, Anwendung
Anwendung lokal erreichbar, remote nicht	Listener-Bindung, Firewall, Routing
Fehler nur zu bestimmten Zeiten	geplanter Job, Last, Lease, Token, Zertifikat
Fehler nach Update	Kompatibilität, Treiber, Richtlinie, Neustart
Fehler verschwindet nach Neustart	Prozesszustand, Cache, Ressourcenleck
Neue Geräte betroffen	DHCP, NAC, Zertifikat, Provisionierung
Alte Geräte betroffen, neue funktionieren	veraltete Konfiguration oder Software

---

8. Belege für und gegen eine Hypothese sammeln

Eine professionelle Hypothese enthält nicht nur unterstützende Hinweise. Es müssen auch widersprechende Beobachtungen gesucht werden.

Hypothese	Unterstützende Hinweise	Widersprechende Hinweise
DNS-Problem	Name wird nicht aufgelöst	Direkte DNS-Abfrage liefert korrekte Adresse

Hypothese	Unterstützende Hinweise	Widersprechende Hinweise
Serverdienst ausgefallen	Port ist von allen Clients geschlossen	Dienst lauscht und andere Clients funktionieren
Benutzerberechtigung fehlt	Nur ein Benutzer betroffen	Derselbe Benutzer funktioniert auf anderem Client
Lokale Firewall blockiert	Nur ein Client betroffen	Verbindung scheitert auch von anderen Clients
VLAN-ACL blockiert	Nur ein VLAN betroffen	Pakete erreichen nachweislich den Server
Zertifikat abgelaufen	TLS-Fehler und Ablaufdatum überschritten	Zertifikat ist gültig und Hostname stimmt
Datenträger voll	Freier Speicher ist nahezu null	Ausreichend Speicher und keine I/O-Fehler
Update verursacht Fehler	Fehler beginnt direkt nach Update	Identisch aktualisierte Systeme funktionieren

> Eine Hypothese sollte herabgestuft werden, wenn mehrere gesicherte Fakten ihr widersprechen.

---

9. Prüfungen sinnvoll priorisieren

Nicht automatisch die wahrscheinlichste Hypothese wird zuerst geprüft. Sinnvoll ist zunächst ein Test, der mit geringem Risiko möglichst viele Hypothesen unterscheidet.

Priorisierungskriterien

Kriterium	Frage
Wahrscheinlichkeit	Passt die Hypothese zum Fehlerbild?
Informationsgewinn	Wie viele Ursachen kann der Test unterscheiden?
Sicherheit	Kann der Test Datenverlust oder Ausfall verursachen?
Aufwand	Wie viel Zeit und Vorbereitung benötigt der Test?
Reproduzierbarkeit	Kann der Test zuverlässig wiederholt werden?
Reversibilität	Lässt sich eine Veränderung sicher zurücknehmen?
Reichweite	Betrifft der Test einen Client oder die Produktion?
Beweislage	Gibt es bereits unterstützende oder widersprechende Fakten?

---

10. Prioritätsklassen

Priorität	Bedeutung	Beispiele
P1	Sicher, schnell und hoher Informationsgewinn	Status, Logs, DNS-, TCP- und Vergleichstest
P2	Gezielte Prüfung mit begrenztem Aufwand	Testkonto, anderer Client, isolierte Konfiguration
P3	Potenziell störende oder aufwendige Prüfung	Dienstneustart, Failover, Rollback
P4	Unwahrscheinlich, widersprochen oder unverhältnismäßig	großflächige Änderung ohne ausreichende Belege

Sinnvolle Reihenfolge

1. passive Zustandsabfrage,
2. Vergleich mit funktionierendem System,
3. gezielte Netzwerk- oder Anwendungstests,
4. Test in isolierter Umgebung,
5. reversible Änderung,
6. Dienstneustart oder Failover,
7. Konfigurationsrollback,
8. weitreichende Änderung oder Produktionsunterbrechung.

---

11. Tests müssen Hypothesen unterscheiden

Ein Test ist besonders nützlich, wenn unterschiedliche Hypothesen unterschiedliche Ergebnisse erwarten lassen.

Wenig aussagekräftiger Test

`Der Benutzer soll es noch einmal versuchen.`

Dieser Test zeigt lediglich, ob der Fehler weiterhin auftritt.

Aussagekräftiger Test

`Derselbe Benutzer testet dieselbe Funktion auf einem funktionierenden Vergleichsclient.`

Mögliche Interpretation:

Ergebnis	Schlussfolgerung
Benutzer funktioniert auf Vergleichsclient	Clientbezogene Ursache wahrscheinlicher
Benutzer funktioniert auch dort nicht	Konto oder Berechtigung wahrscheinlicher
Andere Benutzer funktionieren auf betroffenem Client	Benutzerbezogene Ursache wahrscheinlicher

Ergebnis	Schlussfolgerung
Kein Benutzer funktioniert auf betroffenem Client	Client- oder Netzwerkursache wahrscheinlicher

---

12. Vor dem Test das erwartete Ergebnis festlegen

Vor jedem Test sollte dokumentiert werden:

- welche Hypothese geprüft wird,
- welcher Befehl oder Ablauf verwendet wird,
- welches Ergebnis die Hypothese unterstützt,
- welches Ergebnis sie widerlegt,
- welche Ergebnisse uneindeutig wären,
- ob der Test Veränderungen oder Risiken verursacht,
- wie der Ausgangszustand wiederhergestellt wird.

Testvorlage

Feld	Eintrag
Hypothese	
Technische Begründung	
Test	
Erwartetes Ergebnis	
Widerlegendes Ergebnis	
Risiko	
Rückfallmöglichkeit	
Tatsächliches Ergebnis	
Bewertung	Bestätigt / Gestützt / Unklar / Widerlegt
Nächster Schritt	

---

13. Grundlegender Netzwerk- und Diensttest

Als Beispiel wird ein Webdienst verwendet:

- Hostname: `server.example`
- Port: `443`
- Protokoll: `HTTPS`

`server.example` ist ein reservierter Beispieldomainname und muss durch das tatsächliche Ziel ersetzt werden.

---

14. Stufe 1 – Namensauflösung prüfen

Windows

[TEST] Resolve-DnsName server.example

Bestimmten DNS-Server abfragen

[TEST] Resolve-DnsName server.example -Server <DNS-Server-IP>

Alternative

[TEST] nslookup server.example

Linux

[TEST] getent ahosts server.example

[TEST] dig server.example

Kurze Ausgabe

[TEST] dig +short server.example

Bestimmten DNS-Server abfragen

[TEST] dig @<DNS-Server-IP> server.example

---

macOS

[TEST] dscacheutil -q host -a name server.example

[TEST] dig server.example

Kurze Ausgabe

[TEST] dig +short server.example

Bestimmten DNS-Server abfragen

[TEST] dig @<DNS-Server-IP> server.example

Interpretation der DNS-Prüfung

Ergebnis	Mögliche Bedeutung
Keine Adresse wird geliefert	DNS-Eintrag, DNS-Server oder Resolverproblem

Ergebnis	Mögliche Bedeutung
Falsche IP-Adresse	veralteter Eintrag, falsche Zone, Cache oder Split-DNS
Unterschiedliche Clients erhalten unterschiedliche IPs	Split-DNS, Load Balancing, Cache oder andere Resolver
Bestimmter DNS-Server antwortet korrekt, Standardabfrage nicht	Client verwendet falschen oder nicht erreichbaren DNS-Server
Name wird korrekt aufgelöst	DNS-Grundfunktion wahrscheinlich vorhanden
Auflösung dauert sehr lange	DNS-Server, Weiterleitung, Netzwerk oder Suchdomäne prüfen

Eine erfolgreiche DNS-Abfrage beweist noch nicht, dass der Zielservice erreichbar ist.

---

15. Stufe 2 – IP-Erreichbarkeit mit ICMP prüfen

Windows

`[TEST] ping -n 4 <IP-Adresse>`

Linux

`[TEST] ping -c 4 <IP-Adresse>`

macOS

`[TEST] ping -c 4 <IP-Adresse>`

Interpretation

Ergebnis	Bedeutung
Antworten werden empfangen	IP-Kommunikation per ICMP funktioniert
Keine Antwort	Host, Route, Firewall oder ICMP-Filterung möglich
Hohe Latenz	Netzlast, WAN-Strecke, WLAN oder Zielauslastung möglich
Paketverlust	instabile Verbindung, Überlastung oder physisches Problem möglich
„Destination unreachable“	Route, Gateway oder Zielnetz nicht erreichbar
Ping per IP funktioniert, per Name nicht	Namensauflösung priorisieren

> Ein fehlgeschlagener Ping beweist nicht, dass der Server ausgefallen ist. ICMP kann blockiert sein, während der eigentliche TCP-Dienst funktioniert.

---

16. Stufe 3 – Zielport prüfen

Windows

`[TEST] Test-NetConnection server.example -Port 443 -InformationLevel Detailed`

Wichtige Felder:

- `RemoteAddress`
- `RemotePort`
- `InterfaceAlias`
- `SourceAddress`
- `TcpTestSucceeded`

Nur Ergebnis ausgeben

`[TEST] Test-NetConnection server.example -Port 443 -InformationLevel Quiet`

Linux

`[TEST] nc -vz -w 3 server.example 443`

---

macOS

`[TEST] nc -vz -w 3 server.example 443`

`nc` kann je nach Linux-Installation fehlen und muss gegebenenfalls als freigegebenes Paket installiert werden.

Interpretation des Porttests

Ergebnis	Mögliche Bedeutung
TCP-Verbindung erfolgreich	Netzwerkpfad und TCP-Listener grundsätzlich erreichbar
Verbindung abgelehnt	Ziel erreichbar, aber kein Listener oder aktive Ablehnung
Zeitüberschreitung	Paketfilter, Route, Überlastung oder nicht antwortendes Ziel
Name kann nicht aufgelöst werden	DNS zuerst untersuchen
Nur ein Client scheitert	Client, lokales Netzwerk oder lokale Firewall
Nur ein VLAN scheitert	ACL, Firewall, Routing oder VLAN-Konfiguration
Alle Clients scheitern	Dienst, Server, zentrale Firewall oder gemeinsame Abhängigkeit

Ein erfolgreicher TCP-Test beweist nur den Verbindungsaufbau zum Port. Er bestätigt noch nicht, dass Anmeldung, Anwendung oder Datenbank funktionieren.

---

17. Stufe 4 – Anwendungsebene prüfen

Windows

`[TEST] curl.exe -v -o NUL https://server.example/`

Linux

`[TEST] curl -v -o /dev/null https://server.example/`

macOS

`[TEST] curl -v -o /dev/null https://server.example/`

Die Option `-v` zeigt unter anderem:

- aufgelöste IP-Adresse,
- TCP-Verbindungsaufbau,
- TLS-Aushandlung,
- Zertifikatsprüfung,
- gesendete HTTP-Anfrage,
- empfangenen HTTP-Status.

> Beim ersten Test sollte nicht `-k` beziehungsweise `--insecure` verwendet werden. Diese Option deaktiviert die Zertifikatsprüfung und kann genau den Fehler verbergen, der untersucht werden soll.

---

18. HTTP-Ergebnisse interpretieren

Ergebnis	Interpretation
`200 OK`	Anfrage wurde erfolgreich verarbeitet
`301` oder `302`	Weiterleitung; Ziel des `Location`-Headers prüfen
`401 Unauthorized`	Dienst erreichbar, Authentifizierung erforderlich oder fehlgeschlagen
`403 Forbidden`	Dienst erreichbar, Zugriff wird verweigert
`404 Not Found`	Dienst erreichbar, Ressource oder Pfad nicht gefunden
`408 Request Timeout`	Anfrage wurde nicht rechtzeitig verarbeitet
`429 Too Many Requests`	Rate Limit oder zu viele Anfragen

Ergebnis	Interpretation
`500 Internal Server Error`	serverseitiger Anwendungsfehler
`502 Bad Gateway`	Proxy oder Gateway erhält ungültige Backend-Antwort
`503 Service Unavailable`	Dienst oder Backend nicht verfügbar
`504 Gateway Timeout`	Proxy oder Gateway wartet vergeblich auf Backend
TLS-Zertifikatsfehler	Zertifikat, Hostname, Vertrauenskette oder Uhrzeit prüfen
TCP-Verbindungsfehler	Netzwerkpfad, Firewall oder Listener prüfen

Ein HTTP-Status ist ein wichtiger Hinweis, aber die genaue Bedeutung kann von der Anwendung abweichen.

---

19. TLS-Verbindung untersuchen

Auf Linux und macOS sowie unter Windows mit installiertem OpenSSL:

`[TEST] openssl s_client -connect server.example:443 -servername server.example -showcerts`

Zu prüfen sind:

- Zertifikatsinhaber,
- Subject Alternative Names,
- Aussteller,
- Gültigkeitszeitraum,
- Zertifikatskette,
- verwendete TLS-Version,
- verwendete Cipher Suite,
- abschließendes Verifikationsergebnis.

Typische Befunde

Befund	Mögliche Ursache
Zertifikat abgelaufen	Zertifikat wurde nicht rechtzeitig erneuert
Zertifikat noch nicht gültig	falsche Systemzeit oder falsches Zertifikat
Hostname stimmt nicht	falsches Zertifikat, Alias oder fehlender SAN-Eintrag
Unbekannte CA	Stamm- oder Zwischenzertifikat fehlt
Verbindung ohne Zertifikat beendet	TLS-Listener oder Proxyproblem
Handshake Failure	inkompatible TLS-Version, Cipher oder Clientzertifikat
Browser funktioniert, Anwendung nicht	unterschiedlicher Trust Store oder TLS-Stack

---

20. Netzwerkpfad prüfen

Windows

`[TEST] tracert server.example`

PowerShell

`[TEST] Test-NetConnection server.example -TraceRoute`

Linux

`[TEST] tracepath server.example`

Alternativ, wenn installiert:

`[TEST] traceroute server.example`

---

macOS

`[TEST] traceroute server.example`

Wichtige Einschränkungen

- Nicht jeder Router beantwortet Traceroute-Anfragen.
- Sternchen bedeuten nicht automatisch einen Fehler an diesem Gerät.
- Der Hin- und Rückweg kann unterschiedlich sein.
- Firewalls können Traceroute blockieren.
- Der letzte erreichbare Hop ist nicht automatisch die Fehlerursache.

---

21. Lokal gegen remote testen

Wenn ein Dienst direkt auf dem Server funktioniert, aber von Clients nicht erreichbar ist, wird die Ursache auf Netzwerkpfad, Listener-Bindung oder Firewall eingegrenzt.

Windows-Server

Dienst lokal testen

`[TEST] Test-NetConnection localhost -Port <Port>`

Listener prüfen

`[RO][PRIV] Get-NetTCPConnection -State Listen -LocalPort <Port>`

Zugehörigen Prozess bestimmen

[RO] Get-Process -Id <OwningProcess>

Linux-Server

Dienst lokal testen

[TEST] nc -vz -w 3 localhost <Port>

Listener prüfen

[RO][PRIV] ss -lntp | grep ":<Port>"

HTTP-Dienst lokal prüfen

[TEST] curl -v http://localhost:<Port>/

---

macOS-Server

Dienst lokal testen

[TEST] nc -vz -w 3 localhost <Port>

Listener prüfen

[RO][PRIV] lsof -nP -iTCP:<Port> -sTCP:LISTEN

Interpretation

Lokaler Test	Remote-Test	Wahrscheinlicher Bereich
Erfolgreich	Erfolgreich	Anwendung, Anmeldung oder Benutzerberechtigung
Erfolgreich	Fehlgeschlagen	Firewall, Routing, NAT oder Listener-Bindung
Fehlgeschlagen	Fehlgeschlagen	Dienst, Server oder Anwendung
Fehlgeschlagen	Erfolgreich	ungewöhnlicher Proxy-, Container- oder Load-Balancer-Pfad
Unterschiedlich je Quellnetz	Unterschiedlich	VLAN, ACL, Firewall oder Routing

---

22. Dienstzustand und Protokolle prüfen

Windows

[RO] Get-Service -Name <Dienstname>

[RO] Get-WinEvent -FilterHashtable @{LogName="System"; StartTime=(Get-Date).AddHours(-1)} | Select-Object TimeCreated, Id, ProviderName, LevelDisplayName, Message

---

Linux

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

[RO] journalctl -u <Dienstname> --since "1 hour ago" --no-pager

---

macOS

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

[RO] log show --last 1h --predicate 'process == "<Prozessname>"' --style compact

Zu prüfende Hinweise

- Dienst läuft nicht,
- Dienst startet wiederholt neu,
- Abhängigkeit ist ausgefallen,
- Port kann nicht gebunden werden,
- Konfigurationsdatei enthält Fehler,
- Dienstkonto kann sich nicht anmelden,
- Zertifikat kann nicht geladen werden,
- Datenträger ist voll,
- Datei oder Datenbank ist gesperrt,
- Arbeitsspeicher reicht nicht aus.

---

23. Benutzer- und Berechtigungshypothesen prüfen

Windows

Benutzeridentität und Gruppen

[RO][SENSITIV] whoami /all

NTFS-Berechtigungen

[RO][SENSITIV] icacls "<Pfad>"

PowerShell-ACL

[RO][SENSITIV] Get-Acl "<Pfad>" | Format-List

Domänenkanal auf Mitgliedscomputern prüfen

[TEST] Test-ComputerSecureChannel -Verbose

> `Test-ComputerSecureChannel` ist für Domänenmitgliedscomputer vorgesehen. Auf Domänencontrollern ist dieses Cmdlet laut Microsoft nicht zur zuverlässigen Prüfung geeignet. Der Parameter `-Repair` wäre eine Änderung und gehört nicht zum reinen Test.

Linux

Benutzer und Gruppen

`[RO] id <Benutzername>`

Datei- und Verzeichnisrechte

`[RO] ls -ld "<Pfad>"`

Rechte des vollständigen Pfades

`[RO] namei -l "<Pfad>"`

POSIX-ACL anzeigen

`[RO][SENSITIV] getfacl "<Pfad>"`

---

macOS

Benutzer und Gruppen

`[RO] id <Benutzername>`

Dateirechte und ACL

`[RO][SENSITIV] ls -lde "<Pfad>"`

Typische Vergleichstests

Test	Interpretation
Anderer Benutzer funktioniert am selben Client	Benutzer oder Berechtigung priorisieren
Derselbe Benutzer funktioniert an anderem Client	Client oder Benutzerprofil priorisieren
Testkonto mit gleicher Rolle funktioniert	individuelles Konto untersuchen
Testkonto mit anderer Rolle funktioniert	Rollen- oder Gruppenberechtigung untersuchen
Dateizugriff direkt möglich, Anwendung nicht	Anwendung oder Dienstkonto untersuchen

Bei Anmeldetests muss eine mögliche Kontosperrung berücksichtigt werden.

24. Konfigurationen vergleichen

Windows PowerShell

`[RO] Compare-Object (Get-Content "<Funktionierende-Datei>") (Get-Content "<Betroffene-Datei>")`

Prüfsumme vergleichen

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

Linux und macOS

`[RO] diff -u "<Funktionierende-Datei>" "<Betroffene-Datei>"`

Prüfsumme unter Linux

`[RO] sha256sum "<Datei>"`

Prüfsumme unter macOS

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

Zu vergleichende Werte

- Softwareversion,
- Konfigurationsdatei,
- DNS-Server,
- Proxy,
- Standardgateway,
- Zertifikat,
- Gruppenmitgliedschaft,
- Dateiberechtigung,
- Dienstkonto,
- Firewallprofil,
- Umgebungsvariablen,
- installierte Updates,
- Richtlinien,
- Feature Flags.

> Eine unterschiedliche Prüfsumme zeigt nur, dass Dateien verschieden sind. Sie zeigt nicht, welcher Unterschied relevant ist.

---

25. Diagnosematrix für typische Ergebnisse

Testergebnis	Wahrscheinlicher Fehlerbereich
DNS-Abfrage schlägt fehl	DNS-Client, DNS-Server oder Netzwerk zum DNS-Server
DNS liefert falsche IP	Zone, Cache, Hosts-Datei oder Split-DNS
IP-Ping funktioniert, Name nicht	Namensauflösung
Ping scheitert, TCP funktioniert	ICMP wird wahrscheinlich blockiert
TCP-Port ist erreichbar, HTTP scheitert	Anwendung, TLS, Authentifizierung oder Protokoll
TCP-Verbindung wird abgelehnt	Kein Listener oder aktive Ablehnung
TCP-Verbindung läuft in Timeout	Filterung, Routing, Überlastung oder Ziel antwortet nicht
Lokal funktioniert, remote nicht	Firewall, Routing, Listener-Bindung oder NAT
Andere Clients funktionieren	Clientbezogene Ursache
Andere Benutzer funktionieren	Benutzer, Konto, Profil oder Berechtigung
Gleiches VLAN betroffen	VLAN, ACL, DHCP oder Gateway
Alle VLANs betroffen	Server, Dienst oder zentrale Komponente
HTTP 401	Authentifizierung untersuchen
HTTP 403	Autorisierung oder Richtlinie untersuchen
HTTP 500	Anwendung oder Backend untersuchen
HTTP 502/504	Proxy, Load Balancer oder Backend untersuchen
Zertifikatsfehler	Zeit, Hostname, Trust Store oder Zertifikatskette
Dienst läuft, aber kein Listener	Konfiguration, Bindung oder Startfehler
Listener vorhanden, Anwendung reagiert nicht	Prozesszustand, Abhängigkeit oder Ressourcen
Datenträger voll	Speicherverbrauch und Logwachstum untersuchen

---

26. Mehrdeutige Ergebnisse erkennen

Ein einzelner Test liefert häufig keine eindeutige Ursache.

Beispiel: TCP-Timeout

Mögliche Ursachen:

- lokale Firewall,
- zentrale Firewall,
- fehlende Route,
- falsches Gateway,
- Zielsystem ausgeschaltet,
- Dienst überlastet,
- asymmetrisches Routing,

- falsche IP-Adresse,
- Sicherheitsfilter verwirft Pakete,
- Providerproblem.

Deshalb folgt auf einen mehrdeutigen Test ein zweiter Test, der diese Möglichkeiten weiter trennt.

Geeignete Folgeprüfungen

- Test von anderem Client,
- Test aus anderem VLAN,
- Test direkt auf dem Server,
- Listener auf dem Server prüfen,
- Firewall-Logging prüfen,
- Paketaufzeichnung auf Client oder Server,
- Routingtabellen vergleichen,
- Monitoring und zentrale Protokolle prüfen.

---

27. Beispiel einer vollständigen Hypothesenliste

Störung

Clients aus VLAN 30 können `https://server.example` nicht öffnen. Clients aus VLAN 40 funktionieren. DNS liefert in beiden VLANs dieselbe IP-Adresse.

Nr.	Hypothese	Belege dafür	Belege dagegen	Test	Priorität
H1	ACL blockiert VLAN 30 zu TCP 443	Nur VLAN 30 betroffen	Noch keine	TCP-Test und Firewall-Log	P1
H2	Falsches Gateway in VLAN 30	gesamtes VLAN betroffen	Andere Ziele erreichbar	Route und Gateway prüfen	P1
H3	DNS liefert falsche Adresse	Webdienst nicht erreichbar	Beide VLANs erhalten dieselbe IP	DNS-Ergebnisse vergleichen	P4
H4	Webdienst ist ausgefallen	Verbindung schlägt fehl	VLAN 40 funktioniert	lokaler und externer Porttest	P4
H5	MTU-Problem im Netzwerkpfad	HTTPS betroffen	noch keine Größenabhängigkeit bekannt	Paketgröße und Paketaufzeichnung	P2
H6	Proxy wird nur in VLAN 30 verwendet	nur bestimmte Clients betroffen möglich	noch ungeprüft	Proxykonfiguration vergleichen	P2

Erste Prüfung

`Test-NetConnection server.example -Port 443`

Ergebnis:

- VLAN 30: `TcpTestSucceeded: False`
- VLAN 40: `TcpTestSucceeded: True`

Dadurch werden H1, H2 oder H6 wahrscheinlicher. H3 und H4 werden weniger wahrscheinlich.

Zweite Prüfung

- Standardgateway in VLAN 30 ist korrekt.
- Andere Ziele über dasselbe Gateway funktionieren.
- Firewall protokolliert einen verworfenen Verbindungsversuch von VLAN 30 zu TCP 443.

Damit wird H1 stark gestützt.

Bestätigung

Nach einer freigegebenen Korrektur der betroffenen ACL-Regel funktioniert derselbe definierte Test aus VLAN 30. Die Firewall protokolliert nun eine erlaubte Verbindung.

---

28. Hypothesenstatus verwenden

Status	Bedeutung
Offen	Noch nicht geprüft
Gestützt	Hinweise sprechen dafür
Teilweise gestützt	Einige Ergebnisse passen, andere sind offen
Unklar	Test war nicht eindeutig
Widerlegt	Ergebnis widerspricht der Vorhersage
Bestätigt	Mehrere Belege und Gegenprobe stützen die Ursache
Zurückgestellt	Derzeit geringe Priorität
Nicht prüfbar	Werkzeug, Zugriff oder Vergleichssystem fehlt

Eine Hypothese sollte nicht bereits nach einem passenden Einzelbefund als bestätigt gelten.

---

29. Wann gilt eine Ursache als bestätigt?

Eine Ursache ist ausreichend bestätigt, wenn möglichst mehrere Kriterien erfüllt sind:

- sie erklärt das gesamte Fehlerbild,
- sie erklärt den betroffenen Umfang,

- sie passt zum zeitlichen Verlauf,
- passende technische Belege sind vorhanden,
- alternative Hypothesen wurden widerlegt,
- der Fehler kann reproduziert werden,
- eine Gegenprobe verändert das Ergebnis wie erwartet,
- die Korrektur beseitigt den ursprünglichen Fehler,
- die Korrektur verursacht keine neuen Störungen.

Besonders starke Bestätigung

Zustand	Ergebnis
Fehlerzustand vorhanden	definierter Test schlägt fehl
Ursache kontrolliert entfernt	derselbe Test funktioniert
Ursache in Testumgebung erneut hergestellt	derselbe Test schlägt wieder fehl
Ursache erneut entfernt	derselbe Test funktioniert wieder

In Produktionsumgebungen ist ein erneutes absichtliches Herstellen des Fehlers häufig nicht vertretbar. In diesem Fall müssen Protokolle, Vergleichssysteme und technische Plausibilität gemeinsam bewertet werden.

---

30. Typische Denkfehler vermeiden

Denkfehler	Beschreibung	Gegenmaßnahme
Bestätigungsfehler	Es werden nur Belege für die Lieblingshypothese gesucht	aktiv nach widersprechenden Fakten suchen
Fixierung	Erste Erklärung wird nicht mehr hinterfragt	mindestens mehrere Hypothesen notieren
Zeitliche Verwechslung	Änderung vor Fehler wird automatisch zur Ursache	technische Verbindung und Gegenprobe verlangen
Autoritätseffekt	Aussage einer erfahrenen Person wird ungeprüft übernommen	Messwerte und Protokolle verwenden
Verfügbarkeitseffekt	Kürzlich erlebter Fehler wird überall vermutet	aktuellen Umfang und aktuelle Fakten prüfen
Aktionismus	Sichtbare Aktivität ersetzt systematische Diagnose	erwartetes Testergebnis vorab definieren
Erfolgsfehlschluss	Nach Neustart geht es, daher sei Ursache gefunden	Wiederherstellung und Ursachenbestätigung trennen
Werkzeuggläubigkeit	Ausgabe eines Werkzeugs wird als absolute Wahrheit angesehen	Messgrenzen und Gegenprüfung beachten
Scheingenauigkeit	Punktwert wird mit Gewissheit verwechselt	Priorisierung als Hilfsmittel verwenden

Denkfehler	Beschreibung	Gegenmaßnahme
Gruppendenken	Team übernimmt eine Vermutung ohne Gegenargument	gezielt eine alternative Erklärung verlangen

31. Regeln für aussagekräftige Tests

- Möglichst nur eine Variable verändern.
- Vorher und nachher denselben Test verwenden.
- Erwartetes Ergebnis vorher festlegen.
- Funktionierendes Vergleichssystem verwenden.
- Quell- und Zielsystem dokumentieren.
- Uhrzeit und Zeitzone dokumentieren.
- Exakten Befehl dokumentieren.
- Vollständige Ausgabe sichern.
- Fehlercode nicht nur sinngemäß wiedergeben.
- Test möglichst mehrfach reproduzieren.
- Belastung und Nebenwirkungen berücksichtigen.
- Nach negativer Prüfung Hypothese herabstufen.
- Bei unklarem Ergebnis einen trennschärferen Test wählen.
- Keine produktionsweite Änderung als ersten Test verwenden.

32. Dokumentationsvorlage für Hypothesen

Nr.	Hypothese	Technische Begründung	Belege dafür	Belege dagegen	Geplanter Test	Erwartetes Ergebnis	Risiko	Priorität	Ergebnis	Status
H1										
H2										
H3										
H4										

33. Dokumentationsvorlage für einen Test

Feld	Eintrag
Ticketnummer	
Hypothese	
Testnummer	
Datum und Uhrzeit	
Zeitzone	

Feld	Eintrag
Ausführende Person	
Quellsystem	
Quell-IP	
Zielsystem	
Ziel-IP	
Protokoll und Port	
Ausgangszustand	
Exakter Befehl	
Erwartetes Ergebnis	
Tatsächliches Ergebnis	
Vollständige Ausgabe gesichert	Ja / Nein
Hypothese gestützt	Ja / Nein / Unklar
Nebenwirkungen	
Rückfall erforderlich	Ja / Nein
Nächster Test	

Kurzcheckliste

- [ ] Problembeschreibung präzise formuliert
- [ ] Fakten und Vermutungen getrennt
- [ ] Betroffene und nicht betroffene Systeme berücksichtigt
- [ ] Mehrere plausible Hypothesen aufgestellt
- [ ] Für jede Hypothese technischer Zusammenhang beschrieben
- [ ] Unterstützende Belege notiert
- [ ] Widersprechende Belege notiert
- [ ] Erwartetes Testergebnis vorher definiert
- [ ] Widerlegendes Ergebnis vorher definiert
- [ ] Risiko und Aufwand des Tests bewertet
- [ ] Sichere Tests mit hohem Informationsgewinn priorisiert
- [ ] Funktionierendes Vergleichssystem verwendet
- [ ] Möglichst nur eine Variable verändert
- [ ] Exakten Befehl und Zeitpunkt dokumentiert
- [ ] Testergebnis vollständig gesichert

- [ ] Hypothesenstatus nach jedem Test aktualisiert
- [ ] Alternative Ursachen nicht vorschnell ausgeschlossen
- [ ] Wiederherstellung und Ursachenbestätigung getrennt
- [ ] Ursache durch Gegenprobe oder mehrere Belege bestätigt

Ergebnis dieses Arbeitsschrittes

Am Ende liegt keine unsortierte Sammlung von Vermutungen mehr vor, sondern ein priorisierter Prüfplan.

Jede wichtige Hypothese enthält:

- eine konkrete mögliche Ursache,
- einen technischen Zusammenhang,
- unterstützende und widersprechende Fakten,
- einen möglichst sicheren Test,
- ein erwartetes Ergebnis,
- ein widerlegendes Ergebnis,
- einen dokumentierten Status.

Dadurch wird die Fehlersuche nachvollziehbar, reproduzierbar und wesentlich effizienter.

Nächste Seite:
1.6 Prüfungen kontrolliert durchführen und Ergebnisse bewerten

---

Offizielle Hersteller- und Standarddokumentation

- Microsoft Learn – Test-NetConnection
- Microsoft Learn – Resolve-DnsName
- Microsoft Learn – DNS-Clientprobleme untersuchen
- Microsoft Learn – Test-ComputerSecureChannel
- freedesktop.org – systemctl
- freedesktop.org – journalctl
- curl – Offizielle Befehlsreferenz
- OpenSSL – openssl s_client
- RFC Editor – RFC 9110: HTTP Semantics
- Apple Support – Protokollmeldungen in der Konsole anzeigen
