6.4 Host erreichbar, Dienst nicht – Ports, Firewall und Anwendungen systematisch prüfen

Ein Zielsystem kann per IP-Adresse erreichbar sein, während der benötigte Dienst trotzdem nicht funktioniert.

Typische Beispiele:

- Ping auf den Server funktioniert, die Website öffnet sich jedoch nicht.
- Der Servername wird korrekt aufgelöst, eine SSH-Verbindung scheitert aber.
- Eine Dateifreigabe ist nicht erreichbar, obwohl der Server antwortet.
- Die Anwendung meldet eine Zeitüberschreitung.
- Ein TCP-Port ist erreichbar, die Anwendung liefert dennoch einen Fehler.
- Der Dienst funktioniert lokal auf dem Server, aber nicht von anderen Geräten.
- Einige Clients erreichen den Dienst, andere nicht.
- Eine Verbindung funktioniert intern, über VPN oder Internet jedoch nicht.

Die Erreichbarkeit eines Hosts und die Erreichbarkeit eines Dienstes sind getrennte Prüfungen.

```
Host erreichbar
    ↓
Zielport erreichbar
    ↓
TCP- oder UDP-Kommunikation funktioniert
    ↓
Dienst nimmt Anfragen an
    ↓
Anwendungsprotokoll funktioniert
    ↓
Authentifizierung und Berechtigung funktionieren
```

Ein erfolgreicher Ping bestätigt deshalb weder einen offenen Port noch einen funktionsfähigen Dienst.

---

1. Fehlerbild exakt erfassen

Vor der technischen Prüfung müssen die betroffene Verbindung und der erwartete Dienst eindeutig bestimmt werden.

Zu dokumentieren sind:

- Quellgerät,
- Quell-IP-Adresse,
- Zielname,
- Ziel-IP-Adresse,
- Zielport,
- Transportprotokoll TCP oder UDP,
- verwendete Anwendung,
- Zeitpunkt des Fehlers,
- genaue Fehlermeldung,
- betroffene Benutzer oder Standorte,
- funktionierende Vergleichsverbindungen,
- letzte bekannte Funktionsfähigkeit,
- kürzlich vorgenommene Änderungen.

Beispiel:

```
Quelle: Client-17, 192.0.2.117
Ziel: appserver.example.test, 192.0.2.50
Dienst: HTTPS
Transport: TCP
Zielport: 443
Fehler: Verbindung nach etwa 20 Sekunden abgebrochen
Betroffen: Clients im VLAN 30
Nicht betroffen: Clients im VLAN 20
```

Die Aussage „Der Server geht nicht“ reicht nicht aus. Benötigt wird eine genaue Beschreibung der betroffenen Kommunikationsbeziehung.

---

2. Host-Erreichbarkeit und Dienst-Erreichbarkeit unterscheiden

Ein Ping verwendet ICMP. Anwendungsdienste verwenden dagegen meistens TCP oder UDP.

Beispiel:

```
ping appserver.example.test
```

Ein erfolgreicher Ping kann bestätigen:

- Namensauflösung liefert eine Adresse,
- ICMP-Anfrage erreicht das Ziel,
- ICMP-Antwort gelangt zurück.

Er beweist nicht:

- dass TCP Port 443 geöffnet ist,
- dass eine Firewall den Anwendungsverkehr erlaubt,
- dass der Webserver läuft,
- dass TLS funktioniert,
- dass die Anwendung eine gültige Antwort liefert,
- dass der Benutzer zugriffsberechtigt ist.

Umgekehrt kann ein Dienst funktionieren, obwohl Ping scheitert. ICMP kann gezielt blockiert sein, während der benötigte TCP- oder UDP-Port erlaubt bleibt.

---

3. Den tatsächlich benötigten Port bestimmen

Vor einem Porttest muss bekannt sein, welchen Zielport und welches Transportprotokoll die Anwendung verwendet.

Beispiele:

Dienst	Transport	typischer Port
HTTP	TCP	80
HTTPS	TCP	443
SSH	TCP	22
RDP	TCP und UDP	3389
SMB	TCP	445
DNS	UDP und TCP	53
SMTP	TCP	25
IMAP mit TLS	TCP	993
PostgreSQL	TCP	5432
MySQL	TCP	3306
LDAP	TCP und teilweise UDP	389
LDAPS	TCP	636
NTP	UDP	123
DHCP	UDP	67 und 68

Diese Ports sind typische Standardwerte. Anwendungen können abweichend konfiguriert sein.

Zu prüfen sind daher:

- Herstellerdokumentation,

- Serverkonfiguration,
- Reverse-Proxy-Konfiguration,
- Container-Portzuordnung,
- Load-Balancer-Konfiguration,
- Firewallregeln,
- tatsächlich geöffnete Sockets,
- eventuell verwendete dynamische Ports.

Ein Test des falschen Ports liefert keine verwertbare Aussage über den benötigten Dienst.

---

4. Zieladresse und Zielport gemeinsam prüfen

Ein Port gehört immer zu einer konkreten IP-Adresse und einem Transportprotokoll.

```
192.0.2.50:443/TCP
```

Bei mehreren IP-Adressen muss geprüft werden, welche Adresse die Anwendung tatsächlich verwendet.

Mögliche Fehler:

- DNS liefert eine alte Adresse,
- IPv6 wird gegenüber IPv4 bevorzugt,
- der Dienst lauscht nur auf einer bestimmten Schnittstelle,
- ein Load Balancer verwendet eine andere Adresse,
- NAT leitet auf ein falsches Ziel weiter,
- ein Client verwendet einen Proxy,
- die Anwendung greift auf einen anderen Hostnamen zurück.

Vor der Bewertung eines Porttests sollten deshalb Zielname und aufgelöste Adressen dokumentiert werden.

Windows

```
Resolve-DnsName <zielname>
```

Linux und macOS

```
dig <zielname>
```

---

5. TCP-Port vom Client aus testen

Windows

```
Test-NetConnection <zielname> -Port <port>
```

Kurzform:

```
tnc <zielname> -Port <port>
```

Beispiel:

```
Test-NetConnection appserver.example.test -Port 443
```

Wichtig ist insbesondere:

```
TcpTestSucceeded : True
```

oder:

```
TcpTestSucceeded : False
```

PowerShell 7

```
Test-Connection <zielname> -TcpPort <port>
```

Linux und macOS

```
nc -vz <zielname> <port>
```

Beispiel:

```
nc -vz appserver.example.test 443
```

Alternativ kann bei bestimmten Diensten ein protokollspezifischer Test sinnvoller sein:

```
curl -v http://<zielname>:<port>/
curl -vk https://<zielname>:<port>/
```

> Ein erfolgreicher TCP-Porttest bestätigt zunächst nur, dass eine TCP-Verbindung aufgebaut werden konnte. Er beweist nicht, dass die Anwendung fachlich korrekt arbeitet.

---

6. TCP-Ergebnisse richtig einordnen

Beobachtung	technische Bedeutung	möglicher Untersuchungsbereich
Verbindung erfolgreich	TCP-Verbindungsaufbau abgeschlossen	Anwendung, Protokoll, TLS, Authentifizierung oder Inhalt prüfen
sofort abgelehnt	Ziel oder Zwischenkomponente sendet aktiv eine Ablehnung	kein Listener, falscher Port, Firewall mit Reject oder Dienst beendet
Zeitüberschreitung	keine verwertbare Antwort innerhalb der Wartezeit	Firewall-Drop, Routing, Rückweg, NAT, Zielsystem oder Paketverlust
Verbindung wird sofort zurückgesetzt	TCP-RST beendet die Verbindung	Dienst, Proxy, Firewall, Protokollfehler oder Anwendung
Verbindung beginnt und bleibt hängen	TCP besteht, Anwendung antwortet nicht vollständig	Dienst, Backend, Überlastung, TLS, MTU oder Abhängigkeit
Verbindung funktioniert nur per IP	Namensauflösung, Zertifikat, virtueller Host oder Proxy	
Verbindung funktioniert nur per Name	Anwendung benötigt Hostnamen, SNI oder virtuellen Host	
Port ist offen, Anwendung meldet Fehler	kein reiner Portfehler	Protokoll, Anwendung, Backend, Authentifizierung oder Berechtigung

Eine Zeitüberschreitung beweist nicht automatisch, dass eine Firewall die Verbindung blockiert. Auch ein fehlender Rückweg, ein falsches NAT-Ziel oder ein nicht reagierendes System kann dasselbe Verhalten verursachen.

---

7. TCP-Verbindungsaufbau verstehen

Der normale TCP-Verbindungsaufbau verwendet drei Schritte:

```
Client → Server: SYN
Server → Client: SYN, ACK
Client → Server: ACK
```

Danach kann die Anwendungsübertragung beginnen.

Mögliche Abweichungen:

Paketfolge	mögliche Einordnung
SYN, SYN/ACK, ACK	TCP-Verbindung wurde aufgebaut
wiederholte SYN-Pakete ohne Antwort	Paketverlust, Firewall-Drop, Routing, Ziel oder Rückweg
SYN, danach RST/ACK	Port geschlossen oder Verbindung aktiv abgelehnt
SYN, SYN/ACK, danach kein ACK	Rückweg zum Client, Client-Firewall oder asymmetrischer Pfad
vollständiger Handshake, danach RST	Anwendung, Proxy oder Protokoll beendet die Verbindung
vollständiger Handshake, keine Nutzdatenantwort	Dienst hängt, Backend wartet oder Anwendungsproblem
wiederholte Übertragungen	Paketverlust, Überlastung, MTU oder instabiler Datenpfad

Diese Unterscheidung ist mit einer Paketaufzeichnung möglich.

---

8. UDP-Dienste getrennt betrachten

UDP besitzt keinen TCP-Handshake. Deshalb lässt sich ein UDP-Port nicht genauso zuverlässig wie ein TCP-Port mit einem einfachen Verbindungsversuch bewerten.

Beispiel:

```
nc -vzu <zielname> <port>
```

Ein solcher Test kann Pakete senden. Eine Erfolgsmeldung beweist jedoch nicht zwingend, dass:

- der UDP-Dienst läuft,
- das Paket den Server erreicht,
- der Dienst die Anfrage verarbeitet,
- eine Antwort zurückkommt,
- die Anwendung das richtige Protokoll gesendet hat.

Für UDP sollte möglichst ein protokollspezifisches Werkzeug verwendet werden.

Beispiele:

DNS

```
dig @<dns-server> <name>
```

NTP

```
ntpdate -q <ntp-server>
```

Je nach System können andere Werkzeuge erforderlich sein.

Mögliche UDP-Ergebnisse:

Beobachtung	mögliche Einordnung
gültige Anwendungsantwort	Hin- und Rückweg sowie Dienst grundsätzlich funktionsfähig
ICMP Port Unreachable	am Zielport lauscht wahrscheinlich kein UDP-Dienst
keine Antwort	Dienst antwortet nicht, Firewall, Paketverlust, falsche Anfrage oder Protokollverhalten
Antwort nur bei kleinen Anfragen	Fragmentierung, MTU, EDNS oder Filter
lokal funktioniert, entfernt nicht	Bind-Adresse, Firewall, Routing oder Zugriffsliste

Keine UDP-Antwort ist deshalb weniger eindeutig als eine erfolgreiche oder abgelehnte TCP-Verbindung.

---

9. Auf dem Server prüfen, ob der Dienst lauscht

Ein Dienst muss einen Socket auf der vorgesehenen Adresse und dem vorgesehenen Port geöffnet haben.

Windows

```
Get-NetTCPConnection -State Listen
Get-NetTCPConnection -LocalPort <port>
Get-NetUDPEndpoint -LocalPort <port>
```

Alternativ:

```
netstat -ano
netstat -ano | findstr :<port>
```

Die zugehörige Prozess-ID kann geprüft werden mit:

```
Get-Process -Id <prozess-id>
```

oder:

```
tasklist /FI "PID eq <prozess-id>"
```

Linux

```
ss -lntp
ss -lntp 'sport = :<port>'
ss -lnup
```

Alternativ:

```
sudo lsof -nP -iTCP:<port> -sTCP:LISTEN
sudo lsof -nP -iUDP:<port>
```

macOS

```
sudo lsof -nP -iTCP:<port> -sTCP:LISTEN
sudo lsof -nP -iUDP:<port>
```

Zu prüfen sind:

- ist ein Listener vorhanden?
- ist es der erwartete Prozess?
- verwendet er TCP oder UDP?
- lauscht er auf dem erwarteten Port?
- lauscht er auf der richtigen IP-Adresse?
- verwendet er IPv4, IPv6 oder beides?
- läuft möglicherweise ein anderer Prozess auf dem Port?
- wurde die Konfiguration nach einer Änderung neu geladen?

---

10. Bind-Adresse richtig bewerten

Ein Dienst kann auf unterschiedlichen Adressen lauschen.

Beispiele:

```
127.0.0.1:8080
0.0.0.0:8080
192.0.2.50:8080
[::1]:8080
[::]:8080
```

Typische Bedeutung:

Bind-Adresse	Bedeutung
`127.0.0.1`	nur lokales IPv4-Loopback
`0.0.0.0`	grundsätzlich alle IPv4-Schnittstellen
konkrete IPv4-Adresse	nur diese IPv4-Adresse
`::1`	nur lokales IPv6-Loopback
`::`	grundsätzlich IPv6 und abhängig vom System eventuell zusätzlich IPv4

Ein Dienst, der ausschließlich auf `127.0.0.1` lauscht, kann lokal funktionieren und von entfernten Clients trotzdem nicht erreichbar sein.

Nach Änderungen an Netzwerkschnittstellen oder IP-Adressen kann ein Dienst weiterhin an eine nicht mehr vorhandene oder falsche Adresse gebunden sein.

---

11. Dienststatus und Protokolle prüfen

Ein vorhandener Prozess bedeutet nicht automatisch, dass der Dienst betriebsbereit ist.

Windows

```
Get-Service
Get-Service -Name <dienstname>
```

Linux mit systemd

```
systemctl status <dienstname>
journalctl -u <dienstname>
```

macOS

Je nach Installationsart können `launchctl`, anwendungseigene Werkzeuge oder Protokolldateien verwendet werden.

Zu prüfen sind:

- Dienststatus,
- Startzeitpunkt,
- wiederholte Neustarts,
- Konfigurationsfehler,
- fehlende Zertifikate,
- belegter Port,

- fehlende Berechtigungen,
- nicht erreichbare Datenbank,
- nicht erreichbarer Verzeichnisdienst,
- Speicher- oder Datenträgerprobleme,
- abgelaufene Lizenzen,
- erschöpfte Verbindungen,
- interne Warteschlangen,
- Fehler unmittelbar zum Störungszeitpunkt.

Ein Dienst kann als „running“ angezeigt werden und dennoch keine funktionsfähigen Anfragen bearbeiten.

---

12. Dienst lokal auf dem Server testen

Der lokale Test trennt die Anwendung vom entfernten Netzwerkpfad.

Beispiele:

```
curl -v http://127.0.0.1:<port>/
curl -vk https://127.0.0.1:<port>/
```

Oder:

```
nc -vz 127.0.0.1 <port>
nc -vz <server-ip> <port>
```

Sinnvolle Vergleichstests:

1. Verbindung über `localhost`,
2. Verbindung über Loopback-Adresse,
3. Verbindung über die Server-IP,
4. Verbindung über den vollständigen Namen,
5. Verbindung von einem Client im selben Netz,
6. Verbindung aus dem betroffenen Netz.

Ergebnis	mögliche Einordnung
lokal und entfernt scheitern	Dienst, Listener oder lokale Konfiguration
Loopback funktioniert, Server-IP lokal nicht	Bind-Adresse oder lokale Firewall
lokal funktioniert, entfernt nicht	Firewall, Routing, ACL, NAT oder Netzwerkpfad
IP funktioniert, Name nicht	DNS, virtueller Host, SNI oder Zertifikat
Porttest funktioniert, Protokolltest nicht	Anwendung oder Protokoll

Ergebnis	mögliche Einordnung
ein Netz funktioniert, anderes nicht	Firewallregel, Routing, Segmentierung oder Rückweg

---

13. Lokale Firewall des Servers prüfen

Auch wenn der Dienst lauscht, kann die lokale Firewall eingehende Verbindungen blockieren.

Zu prüfen sind:

- richtige Richtung der Regel,
- richtiges Transportprotokoll,
- richtiger lokaler Port,
- richtige Quellnetze,
- richtiges Netzwerkprofil,
- richtige Anwendung oder Dienstzuordnung,
- IPv4 und IPv6,
- aktive Firewallzone,
- Priorität und Reihenfolge der Regeln,
- protokollierte Ablehnungen,
- zentrale Richtlinien.

Windows

```
Get-NetFirewallProfile
Get-NetFirewallRule -Enabled True
Get-NetFirewallPortFilter
Get-NetFirewallAddressFilter
```

Eine genauere Zuordnung kann über gemeinsame Filter- und Regelobjekte erforderlich sein.

Linux mit firewalld

```
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --list-all
```

Linux mit nftables

```
sudo nft list ruleset
```

Linux mit iptables

```
sudo iptables -L -n -v
sudo ip6tables -L -n -v
```

Firewallregeln sollten nicht pauschal deaktiviert werden. Eine kontrollierte, auf Quelle, Ziel, Port und Protokoll begrenzte Prüfung ist vorzuziehen.

---

14. Client-Firewall und Ausgangsregeln

Nicht nur der Server kann Verbindungen blockieren. Auch auf dem Client können ausgehende Regeln, Sicherheitssoftware oder Endpoint-Richtlinien wirken.

Mögliche Ursachen:

- ausgehender Zielport gesperrt,
- Anwendung darf keine Verbindung herstellen,
- Netzwerkprofil wurde geändert,
- Sicherheitssoftware blockiert den Prozess,
- TLS-Inspection greift ein,
- lokale Proxyrichtlinie erzwingt einen anderen Pfad,
- Host-Firewall verwirft Rückpakete,
- EDR-System beendet die Verbindung.

Wenn andere Anwendungen denselben Zielport erreichen, die betroffene Anwendung aber nicht, sollte zusätzlich eine prozessbezogene Einschränkung untersucht werden.

---

15. Netzwerk-Firewalls und ACLs

Zwischen Client und Server können mehrere Kontrollstellen liegen:

```
Client
  ↓
Client-Firewall
  ↓
VLAN- oder Router-ACL
  ↓
zentrale Firewall
  ↓
VPN-Gateway oder NAT
  ↓
Server-Firewall
  ↓
```

```
Dienst
```

Zu prüfen sind:

- Quell-IP-Adresse,
- Ziel-IP-Adresse,
- Zielport,
- Transportprotokoll,
- Sicherheitszone,
- Regelreihenfolge,
- Zeitplan der Regel,
- Benutzer- oder Anwendungsabhängigkeit,
- NAT vor oder nach der Regelprüfung,
- Sitzungsstatus,
- Trefferzähler,
- Protokolleinträge,
- Rückweg.

Eine Regel für TCP Port 443 erlaubt nicht automatisch UDP Port 443. Ebenso erlaubt eine Regel für IPv4 nicht zwingend IPv6-Verkehr.

---

16. Stateful Firewall und Rückverkehr

Stateful Firewalls verfolgen den Zustand einer Verbindung. Bei einer erlaubten ausgehenden TCP-Verbindung wird der passende Rückverkehr normalerweise über den Verbindungszustand zugeordnet.

Fehler können entstehen durch:

- asymmetrisches Routing,
- Rückpakete passieren eine andere Firewall,
- Firewallzustand ist abgelaufen,
- NAT-Zuordnung fehlt,
- Clusterknoten teilen Zustände nicht korrekt,
- Quelladresse wird unerwartet geändert,
- Verbindung bleibt länger inaktiv als der Session-Timeout,
- Rückverkehr entspricht nicht der gespeicherten Sitzung.

Typisches Symptom:

```
Client sendet SYN.
Server antwortet mit SYN/ACK.
Client erhält die Antwort nicht.
```

In diesem Fall ist der Dienst erreichbar, aber der Rückweg oder eine zustandsabhängige Verarbeitung ist gestört.

---

17. Routing und Rückweg prüfen

Ein erreichbarer Host über ICMP bedeutet nicht, dass jeder Anwendungsverkehr denselben Pfad nimmt oder gleich behandelt wird.

Windows

```
tracert <ziel>
Test-NetConnection <ziel> -TraceRoute
Get-NetRoute
```

Linux

```
traceroute <ziel>
tracepath <ziel>
ip route
ip route get <ziel>
```

macOS

```
traceroute <ziel>
netstat -rn
route -n get <ziel>
```

Zu prüfen sind:

- Route vom Client zum Server,
- Rückroute vom Server zum Client,
- Quell-IP-Auswahl,
- mehrere Standardgateways,
- Policy-Based Routing,
- VPN-Routen,
- überlappende Netze,
- asymmetrische Wege,
- segmentabhängige Firewallregeln.

Die Rückroute muss vom Server oder seinem Gateway zum tatsächlichen Quellnetz führen.

---

18. NAT und Portweiterleitung

Bei Verbindungen über Netzgrenzen kann NAT die Ziel- oder Quelladresse verändern.

Beispiel:

```
öffentliche Adresse:
198.51.100.20:443

NAT-Ziel:
192.0.2.50:8443
```

Zu prüfen sind:

- richtige öffentliche Zieladresse,
- richtiger externer Port,
- richtiges internes Ziel,
- richtiger interner Port,
- TCP oder UDP,
- Quell-NAT,
- Rückroute,
- Firewallregel passend zur NAT-Verarbeitung,
- konkurrierende Portweiterleitung,
- Hairpin-NAT bei Zugriff von intern,
- Ablauf der NAT-Sitzung.

Ein lokal erreichbarer Dienst beweist nicht, dass die Portweiterleitung korrekt ist.

---

19. Container und virtuelle Maschinen

Bei containerisierten oder virtualisierten Diensten existieren zusätzliche Netzwerkschichten.

Mögliche Fehler:

- Container läuft, Port wurde aber nicht veröffentlicht,
- Host-Port zeigt auf den falschen Container-Port,
- Dienst lauscht im Container nur auf Loopback,
- Container-Netzwerk fehlt,
- Firewall auf dem Host blockiert,
- Reverse Proxy erreicht den Container nicht,
- falscher DNS-Name im internen Netzwerk,
- virtuelle Maschine besitzt eine andere IP-Adresse,
- Sicherheitsgruppe erlaubt den Port nicht,
- Dienst wurde auf einen anderen Host verschoben.

Docker

```
docker ps
docker port <container>
docker inspect <container>
```

Beispiel einer Portzuordnung:

```
0.0.0.0:8443 → Container:443/TCP
```

Ein Test auf Host-Port 443 wäre in diesem Beispiel ungeeignet, wenn der Dienst ausschließlich über Host-Port 8443 veröffentlicht wurde.

---

20. Reverse Proxy und Load Balancer

Zwischen Client und Anwendung können Reverse Proxy oder Load Balancer liegen.

```
Client
  ↓
Reverse Proxy oder Load Balancer
  ↓
Backend-Anwendung
```

Mögliche Fehlerbilder:

- Frontend-Port ist erreichbar, Backend jedoch nicht,
- falscher Hostname wird weitergeleitet,
- Health Check schlägt fehl,
- Backend-Port wurde geändert,
- Zertifikat passt nicht,
- virtuelle Hostkonfiguration fehlt,
- Proxy-Zeitüberschreitung ist zu kurz,
- ein einzelner Backend-Server ist fehlerhaft,
- Sitzungspersistenz führt zu einem defekten Knoten,
- Proxy liefert `502`, `503` oder `504`.

Typische HTTP-Ergebnisse:

Ergebnis	mögliche Einordnung
`200`	Anfrage grundsätzlich erfolgreich

Ergebnis	mögliche Einordnung
`301` oder `302`	Weiterleitung prüfen
`400`	Anfrage oder Hostheader ungeeignet
`401`	Authentifizierung erforderlich oder fehlgeschlagen
`403`	Zugriff durch Anwendung oder Richtlinie verweigert
`404`	Pfad, virtueller Host oder Routingregel
`502`	Proxy erhält keine gültige Backend-Antwort
`503`	Dienst oder Backend nicht verfügbar
`504`	Zeitüberschreitung zum Backend

Ein erreichbarer Proxy-Port beweist nicht, dass das Backend funktioniert.

---

21. TLS und Zertifikate prüfen

Bei verschlüsselten Diensten folgt nach dem TCP-Verbindungsaufbau der TLS-Handshake.

Mögliche Fehler:

- Zertifikat abgelaufen,
- Zertifikatsname passt nicht zum Hostnamen,
- Zertifikatskette unvollständig,
- unbekannte Zertifizierungsstelle,
- Client- und Serverprotokolle nicht kompatibel,
- keine gemeinsame Cipher Suite,
- Server Name Indication fehlt oder ist falsch,
- Clientzertifikat erforderlich,
- TLS-Inspection verändert die Verbindung,
- Systemzeit ist falsch.

curl

```
curl -vk https://<zielname>:<port>/
```

OpenSSL

```
openssl s_client -connect <zielname>:<port> -servername <zielname>
```

Dabei können geprüft werden:

- ausgehandeltes TLS-Protokoll,

- präsentierte Zertifikatskette,
- Zertifikatsname,
- Gültigkeitszeitraum,
- Serverantwort,
- Abbruchstelle.

Die Option `-k` bei `curl` überspringt die Zertifikatsprüfung und darf nur zur Eingrenzung verwendet werden. Sie ist keine dauerhafte Lösung für Zertifikatsfehler.

---

22. Hostname, SNI und virtuelle Hosts

Mehrere Webanwendungen können dieselbe IP-Adresse und denselben Port verwenden. Der richtige Dienst wird dann anhand des Hostnamens ausgewählt.

Beispiel:

```
portal.example.test → 192.0.2.50
wiki.example.test   → 192.0.2.50
```

Ein Test ausschließlich gegen die IP-Adresse kann deshalb:

- die falsche Website liefern,
- einen Zertifikatsfehler erzeugen,
- die Standardseite des Proxys anzeigen,
- mit einem HTTP-Fehler enden,
- den falschen virtuellen Host verwenden.

Geeigneter Test:

```
curl -vk https://portal.example.test/
```

Soll eine bestimmte IP-Adresse mit dem korrekten Hostnamen getestet werden:

```
curl -vk --resolve portal.example.test:443:192.0.2.50 https://portal.example.test/
```

Damit werden DNS, Ziel-IP, Hostheader und SNI kontrolliert voneinander getrennt.

---

23. Anwendung und Abhängigkeiten prüfen

Ein Dienst kann den Port öffnen, aber intern von weiteren Komponenten abhängig sein.

Beispiele:

- Datenbank,
- Verzeichnisdienst,
- DNS,
- Dateispeicher,
- Nachrichtenwarteschlange,
- API,
- Lizenzserver,
- externer Identitätsanbieter,
- Zertifikatsdienst,
- weiterer Microservice.

Mögliche Symptome:

- TCP-Verbindung wird aufgebaut,
- Loginseite erscheint,
- Anmeldung scheitert,
- Anfrage endet mit `500`,
- Antwort dauert sehr lange,
- nur bestimmte Funktionen sind betroffen,
- Health Check meldet Fehler,
- Dienst startet, wird aber nicht „ready“.

Die Prüfung muss deshalb vom Frontend bis zu den tatsächlich benötigten Abhängigkeiten fortgesetzt werden.

---

24. Authentifizierung und Berechtigung abgrenzen

Eine erfolgreiche Netzwerkverbindung beweist nicht, dass der Benutzer oder Client den Dienst verwenden darf.

Zu unterscheiden sind:

- Zielport nicht erreichbar,
- TLS-Verbindung fehlerhaft,
- Benutzer nicht authentifiziert,
- falsche Anmeldedaten,
- Konto gesperrt,
- fehlende Gruppenmitgliedschaft,
- Zugriff durch Anwendung verweigert,
- Zugriff durch Netzwerkregel verweigert,
- Kerberos- oder Zertifikatsfehler,
- Mandanten- oder Rollenproblem.

Beispiele:

```
Connection timed out
```

weist eher auf Transport oder fehlende Antwort hin.

```
401 Unauthorized
```

zeigt dagegen, dass eine HTTP-Antwort des Dienstes empfangen wurde und die Untersuchung auf Anwendungsebene fortgesetzt werden muss.

---

25. IPv4 und IPv6 getrennt prüfen

Ein Dienst kann für IPv4 und IPv6 unterschiedlich erreichbar sein.

Windows

```
Resolve-DnsName <zielname> -Type A
Resolve-DnsName <zielname> -Type AAAA
Test-NetConnection <ipv4-adresse> -Port <port>
Test-NetConnection <ipv6-adresse> -Port <port>
```

Linux und macOS

```
curl -4 -v https://<zielname>/
curl -6 -v https://<zielname>/
nc -4 -vz <zielname> <port>
nc -6 -vz <zielname> <port>
```

Mögliche Ursachen:

- Dienst lauscht nur auf IPv4,
- Dienst lauscht nur auf IPv6,
- AAAA-Record zeigt auf eine falsche Adresse,
- IPv6-Firewallregel fehlt,
- IPv6-Rückroute fehlt,
- Client bevorzugt einen nicht funktionsfähigen IPv6-Pfad.

IPv6 sollte nicht pauschal deaktiviert werden. Zuerst ist nachzuweisen, welche Adressfamilie den Fehler verursacht.

---

26. MTU, Fragmentierung und große Datenmengen

Ein TCP-Handshake kann funktionieren, während größere Datenübertragungen scheitern.

Mögliche Symptome:

- Porttest erfolgreich,
- kleine Anfragen funktionieren,
- TLS-Handshake bleibt hängen,
- Dateiübertragung bricht ab,
- bestimmte Webseiten laden nur teilweise,
- Verbindung über VPN scheitert bei größeren Paketen,
- wiederholte TCP-Übertragungen treten auf.

Mögliche Ursachen:

- ungeeignete MTU,
- blockierte ICMP-Meldungen für Path MTU Discovery,
- Fragmentierung,
- Tunnel-Overhead,
- fehlerhafte MSS-Anpassung,
- Paketverlust,
- Sicherheitskomponente verwirft größere Pakete.

Linux

```
tracepath <ziel>
```

Windows

```
ping <ziel> -f -l <paketgröße>
```

Der verwendbare Wert hängt von Protokoll-Headern und Datenpfad ab. Ein einzelner Pingwert darf deshalb nicht ungeprüft als endgültige MTU übernommen werden.

---

27. Proxys und anwendungsspezifische Netzwerkwege

Eine Anwendung kann einen anderen Netzwerkpfad verwenden als ein einfacher Porttest.

Mögliche Unterschiede:

- HTTP-Proxy,
- SOCKS-Proxy,
- automatische Proxykonfiguration,
- VPN-Tunnel,
- DNS over HTTPS,

- anwendungseigener Resolver,
- Service Mesh,
- TLS-Inspection,
- Cloud-Gateway,
- Zero-Trust-Zugriff.

Zu prüfen sind:

- verwendet die Anwendung einen Proxy?
- umgeht das Testwerkzeug den Proxy?
- wird der Zielname durch den Proxy aufgelöst?
- ist der Proxy selbst erreichbar?
- erlaubt der Proxy den Zielport?
- existiert eine Ausnahmeliste?
- wird die Verbindung durch Benutzer- oder Gerätestatus gesteuert?

Ein erfolgreicher direkter Test beweist nicht, dass der von der Anwendung verwendete Proxyweg funktioniert.

---

28. Sporadische und lastabhängige Fehler

Nicht jeder Dienstfehler besteht dauerhaft.

Mögliche Ursachen:

- Verbindungsgrenze erreicht,
- Quellports erschöpft,
- überlasteter Server,
- Load Balancer verteilt auf einen defekten Knoten,
- Firewall-Sitzungstabelle ausgelastet,
- Paketverlust,
- Dienst startet wiederholt neu,
- Datenbankverbindungen erschöpft,
- Timeouts zwischen Proxy und Backend,
- DNS liefert wechselnde Ziele,
- automatische Skalierung reagiert zu langsam.

Windows

```
Get-NetTCPConnection
Get-Counter '\TCPv4\Connections Established'
```

Linux

```
ss -s
ss -ant
```

Wiederholte Tests sollten mit Zeitstempel durchgeführt werden.

PowerShell

```
1..10 | ForEach-Object {
    Get-Date
    Test-NetConnection <ziel> -Port <port>
    Start-Sleep -Seconds 2
}
```

Bash

```
for i in {1..10}; do
    date
    nc -vz <ziel> <port>
    sleep 2
done
```

Zu vergleichen sind:

- Erfolg oder Fehler,
- Antwortzeit,
- Zieladresse,
- betroffener Backend-Server,
- Zeitpunkt,
- gleichzeitige Last,
- Firewall- und Anwendungsprotokolle.

---

29. Paketaufzeichnung einsetzen

Eine Paketaufzeichnung kann zeigen:

- ob der Client ein SYN sendet,
- ob das Ziel mit SYN/ACK oder RST antwortet,
- ob Pakete erneut übertragen werden,
- ob die Verbindung nach dem Handshake abbricht,
- ob TLS beginnt,
- ob eine Anwendungsantwort zurückkommt,

- ob ICMP-Fehlermeldungen auftreten,
- ob IPv4 oder IPv6 verwendet wird,
- ob die tatsächliche Zieladresse stimmt,
- ob Hin- und Rückweg vollständig sind.

Wireshark-Filter

```
ip.addr == <ziel-ip>
tcp.port == <port>
udp.port == <port>
tcp.flags.syn == 1
tcp.flags.reset == 1
tcp.analysis.retransmission
icmp
icmpv6
tls
http
```

Für eine einzelne TCP-Verbindung kann nach Auswahl eines Pakets zusätzlich der zugehörige TCP-Stream gefiltert werden.

Typische Beobachtungen

Paketaufzeichnung	mögliche Einordnung
kein Verbindungsversuch sichtbar	Anwendung, Proxy, Cache oder falsche Schnittstelle
SYN verlässt Client, keine Antwort	Firewall-Drop, Routing, Ziel oder Rückweg
SYN wird mit RST beantwortet	kein Listener oder aktive Ablehnung
SYN/ACK erreicht Client, Client antwortet nicht	Client-Firewall, lokaler Stack oder falscher Zustand
TCP-Handshake vollständig, danach TLS-Fehler	Zertifikat, SNI, Protokoll oder Cipher
TCP-Handshake vollständig, HTTP `401`	Authentifizierung
TCP-Handshake vollständig, HTTP `403`	Berechtigung oder Richtlinie
TCP-Handshake vollständig, HTTP `502`	Proxy- oder Backend-Problem
viele Wiederholungen	Paketverlust, Überlastung, MTU oder Datenpfad
Verbindung wird durch RST beendet	Anwendung, Firewall, Proxy oder Betriebssystem

Paketaufzeichnungen dürfen nur mit entsprechender Berechtigung erstellt und müssen geschützt gespeichert werden.

---

30. Vergleichstests systematisch verwenden

Vergleichstests helfen, die fehlerhafte Grenze einzugrenzen.

Sinnvolle Vergleiche:

Vergleich	mögliche Erkenntnis
betroffener und funktionierender Client	Clientkonfiguration oder Netzsegment
Zielname und Ziel-IP	DNS, SNI oder virtueller Host
IPv4 und IPv6	Adressfamilie oder Datenpfad
lokaler und entfernter Zugriff	Dienst oder Netzwerk
gleiches VLAN und anderes VLAN	Routing, ACL oder Firewall
internes Netz und VPN	VPN-Regel, Route oder DNS
Frontend und Backend direkt	Proxy oder Load Balancer
erster und zweiter Server	einzelnes Zielsystem oder zentraler Pfad
kleiner und großer Datentransfer	MTU, Paketverlust oder Überlastung
TCP-Porttest und Protokolltest	Transport oder Anwendung

Dabei sollte jeweils nur eine Variable verändert werden.

---

31. Praxisfall A: Ping funktioniert, HTTPS-Port nicht

Symptom

- Server antwortet auf Ping,
- Name wird korrekt aufgelöst,
- TCP Port 443 läuft in eine Zeitüberschreitung,
- andere Server im selben Netz sind erreichbar.

Prüfung

1. Zieladresse dokumentieren.
2. TCP Port 443 vom Client testen.
3. Listener auf dem Server prüfen.
4. lokalen HTTPS-Test auf dem Server durchführen.
5. Server-Firewall kontrollieren.
6. Netzwerk-Firewall und Trefferprotokolle prüfen.
7. Paketaufzeichnung auf Client und Server vergleichen.

Mögliche Ursache

Der Webserver läuft und lauscht auf Port 443. Nach einer Netzsegmentierung fehlt jedoch die Freigabe vom neuen Client-VLAN zum Server.

Nachprüfung

- TCP Port 443 ist aus dem vorgesehenen VLAN erreichbar,
- HTTPS liefert eine gültige Antwort,
- nicht vorgesehene Quellnetze bleiben gesperrt,
- Firewallregel ist dokumentiert,
- ursprüngliche Anwendung funktioniert.

---

32. Praxisfall B: Dienst funktioniert lokal, aber nicht entfernt

Symptom

- lokaler Aufruf über `localhost` funktioniert,
- entfernter Porttest wird abgelehnt,
- der Prozess läuft,
- auf dem Server ist kein allgemeiner Listener sichtbar.

Prüfung

1. Listener und Bind-Adresse feststellen.
2. Loopback und Server-IP lokal vergleichen.
3. Dienstkonfiguration prüfen.
4. nach einer Änderung den Dienst kontrolliert neu laden.
5. lokale Firewall berücksichtigen.
6. entfernten Test wiederholen.

Mögliche Ursache

Der Dienst lauscht ausschließlich auf `127.0.0.1`.

Nachprüfung

- Dienst lauscht auf der vorgesehenen Serveradresse,
- lokaler Zugriff funktioniert,
- entfernter Zugriff aus erlaubten Netzen funktioniert,
- Firewall begrenzt den Zugriff weiterhin,
- keine unnötige öffentliche Freigabe wurde geschaffen.

---

33. Praxisfall C: Port 443 ist offen, Website liefert 502

Symptom

- TCP Port 443 ist erreichbar,
- TLS-Verbindung wird aufgebaut,
- Reverse Proxy antwortet mit `502 Bad Gateway`,

- Backend-Anwendung ist nicht direkt erreichbar.

Prüfung

1. Proxy-Antwort dokumentieren.
2. Proxy-Protokolle prüfen.
3. Backend-Ziel und Backend-Port feststellen.
4. Verbindung vom Proxy zum Backend testen.
5. Listener der Backend-Anwendung prüfen.
6. Container- oder Dienststatus kontrollieren.
7. Namensauflösung innerhalb des Proxy-Netzes prüfen.

Mögliche Ursache

Nach einer Änderung verwendet die Backend-Anwendung einen neuen Port. Die Proxy-Konfiguration zeigt weiterhin auf den alten Port.

Nachprüfung

- Proxy erreicht das richtige Backend,
- Backend-Health-Check ist erfolgreich,
- HTTPS-Aufruf liefert die erwartete Anwendung,
- weitere virtuelle Hosts funktionieren,
- Konfigurationsänderung ist dokumentiert.

---

34. Praxisfall D: Einige Clients erreichen den Dienst, andere nicht

Symptom

- Clients im VLAN 20 können eine Anwendung öffnen,
- Clients im VLAN 30 erhalten eine Zeitüberschreitung,
- beide Gruppen lösen denselben Namen zur selben IP-Adresse auf,
- der Dienst funktioniert lokal.

Prüfung

1. Quell-IP-Adressen beider Clients erfassen.
2. Route und Gateway vergleichen.
3. Porttest aus beiden Netzen durchführen.
4. Firewallregeln und Trefferzähler prüfen.
5. Server-Firewall auf Quellnetzbegrenzungen untersuchen.
6. Paketaufzeichnung am Server durchführen.

Mögliche Ursache

Die Server-Firewall erlaubt Port 8443 nur aus dem alten VLAN 20.

Nachprüfung

- vorgesehene Clients aus VLAN 30 erreichen den Dienst,
- VLAN 20 funktioniert weiterhin,
- andere Netze bleiben gesperrt,
- Quellnetze in der Regel sind korrekt dokumentiert.

---

35. Praxisfall E: TCP-Verbindung funktioniert, Anmeldung scheitert

Symptom

- Zielname wird korrekt aufgelöst,
- TCP-Port ist erreichbar,
- TLS funktioniert,
- Anwendung zeigt eine Anmeldeseite,
- Benutzer erhält `401 Unauthorized`.

Prüfung

1. erfolgreichen Transport dokumentieren.
2. Anwendungsprotokoll und Statuscode auswerten.
3. Benutzerkonto und Authentifizierungsquelle prüfen.
4. Uhrzeit, Zertifikate und gegebenenfalls Kerberos berücksichtigen.
5. funktionierendes Vergleichskonto verwenden.
6. Anwendungs- und Authentifizierungsprotokolle prüfen.

Mögliche Ursache

Der Benutzer ist nicht mehr Mitglied der für die Anwendung berechtigten Gruppe.

Nachprüfung

- Benutzer kann sich nach genehmigter Berechtigungskorrektur anmelden,
- TCP- und TLS-Verbindung bleiben unverändert,
- Zugriff ist auf vorgesehene Benutzer begrenzt,
- Ursache und Berechtigungsänderung sind dokumentiert.

---

36. Ungeeignete Sofortmaßnahmen

Problematisch sind insbesondere:

- Firewall vollständig deaktivieren,
- beliebige Ports pauschal freigeben,
- einen Dienst ungeprüft neu starten,
- Server neu starten, ohne den Zustand zu dokumentieren,

- Anwendung als Ursache festlegen, nur weil Ping funktioniert,
- Netzwerk als Ursache festlegen, obwohl der Dienst keinen Listener besitzt,
- Zertifikatsprüfung dauerhaft deaktivieren,
- IPv6 pauschal abschalten,
- NAT-Regeln ohne Sicherung verändern,
- mehrere Firewalls gleichzeitig ändern,
- Sicherheitssoftware ungeprüft entfernen,
- Container neu erstellen, ohne Volumes und Konfiguration zu prüfen,
- produktive Paketaufzeichnungen ungeschützt speichern,
- Testzugänge oder temporäre Freigaben dauerhaft bestehen lassen.

Vor Änderungen sollten dokumentiert werden:

- Quell- und Zieladresse,
- Zielport und Transportprotokoll,
- DNS-Ergebnis,
- Porttestergebnis,
- Listener,
- Dienststatus,
- Firewallregeln,
- NAT-Regeln,
- Routen,
- Protokolle,
- Paketbeobachtungen,
- Zeitpunkt und Umfang der Störung.

---

37. Vollständige Prüfreihenfolge

1. betroffene Anwendung und genaue Fehlermeldung erfassen.
2. Quelle, Zielname und Ziel-IP-Adresse dokumentieren.
3. benötigten Zielport bestimmen.
4. TCP oder UDP unterscheiden.
5. Namensauflösung prüfen.
6. Host-Erreichbarkeit nur als Teilprüfung verwenden.
7. Zielport vom betroffenen Client testen.
8. Ergebnis als Erfolg, Ablehnung, Reset oder Zeitüberschreitung einordnen.
9. funktionierenden Vergleichsclient prüfen.
10. IPv4 und IPv6 getrennt betrachten.
11. auf dem Server den Listener prüfen.
12. Prozess zum Listener bestimmen.
13. Bind-Adresse kontrollieren.
14. Dienststatus und Protokolle auswerten.
15. Dienst über Loopback lokal testen.
16. Dienst über die Server-IP lokal testen.
17. protokollspezifischen Test durchführen.

18. lokale Server-Firewall prüfen.
19. Client-Firewall und Sicherheitssoftware berücksichtigen.
20. Netzwerk-Firewalls und ACLs prüfen.
21. Route vom Client zum Server untersuchen.
22. Rückweg vom Server zum Client prüfen.
23. NAT und Portweiterleitungen berücksichtigen.
24. VPN- und Segmentierungsregeln prüfen.
25. Container-, VM- oder Host-Portzuordnungen prüfen.
26. Reverse Proxy oder Load Balancer untersuchen.
27. Backend-Verbindungen testen.
28. TLS, Zertifikat, SNI und Hostnamen prüfen.
29. Anwendung, Authentifizierung und Berechtigung abgrenzen.
30. bei sporadischen Fehlern wiederholte Tests durchführen.
31. bei Bedarf autorisierte Paketaufzeichnung erstellen.
32. konkrete Hypothese formulieren.
33. genau eine kontrollierbare Änderung durchführen.
34. Listener und Dienststatus erneut prüfen.
35. Porttest wiederholen.
36. protokollspezifischen Test wiederholen.
37. ursprüngliche Anwendung erneut testen.
38. funktionierende und nicht vorgesehene Netze kontrollieren.
39. temporäre Freigaben entfernen.
40. Ursache, Maßnahme und Ergebnis dokumentieren.
41. Monitoring oder Präventionsmaßnahme festlegen.

---

38. Dokumentationsbeispiel

```
Ticket: INC-20644
Beginn: 02.08.2026, etwa 14:20 Uhr
Umfang: Clients im VLAN 30
Betroffen: HTTPS-Anwendung auf appserver.example.test
Nicht betroffen: Clients im VLAN 20

Verbindung:
Quelle: 192.0.2.117
Ziel: appserver.example.test
Ziel-IP: 192.0.2.50
Transport: TCP
Zielport: 443

Symptom:
Der Server antwortet auf ICMP.
```

Die Namensauflösung liefert die korrekte IP-Adresse.
Der HTTPS-Aufruf läuft aus VLAN 30 in eine Zeitüberschreitung.

Ausgangszustand:
- DNS-Auflösung korrekt
- ICMP-Antwort vorhanden
- TCP Port 443 aus VLAN 30 nicht erreichbar
- TCP Port 443 aus VLAN 20 erreichbar
- Webserver lauscht auf 0.0.0.0:443
- lokaler HTTPS-Test auf dem Server erfolgreich
- Server-Firewall erlaubt beide internen Netze
- zentrale Firewall protokolliert verworfene SYN-Pakete aus VLAN 30

Hypothese:
Bei der Einrichtung des neuen VLANs wurde die Freigabe zu TCP Port 443
auf dem Anwendungsserver nicht ergänzt.

Prüfung:
- Quell-IP und Ziel-IP bestätigt
- Route zum Server vorhanden
- Listener und Dienststatus korrekt
- funktionierenden Client aus VLAN 20 verglichen
- Firewallregel enthält nur das Quellnetz von VLAN 20
- keine SYN-Pakete aus VLAN 30 erreichen den Server

Ursache:
Die zentrale Firewallregel erlaubte den HTTPS-Zugriff nur aus VLAN 20.

Maßnahme:
Das freigegebene Quellobjekt wurde nach Genehmigung um das vorgesehene
Clientnetz aus VLAN 30 ergänzt. Ziel und Port blieben unverändert.

Nachprüfung:
- TCP Port 443 aus VLAN 30 erreichbar
- TLS-Handshake erfolgreich
- Anwendung liefert HTTP 200
- Anmeldung funktioniert
- VLAN 20 funktioniert weiterhin
- Zugriff aus nicht freigegebenen Netzen bleibt blockiert

- Firewallprotokoll und Trefferzähler geprüft

Prävention:

- Firewallfreigaben in die VLAN-Abnahme aufnehmen
- automatisierten HTTPS-Health-Check aus beiden Clientnetzen einrichten
- Änderungen an Netzsegmenten mit einer Kommunikationsmatrix prüfen

---

39. Checkliste „Host erreichbar, Dienst nicht“

- [ ] die betroffene Anwendung wurde eindeutig benannt.
- [ ] die genaue Fehlermeldung wurde dokumentiert.
- [ ] Quellgerät und Quell-IP-Adresse wurden erfasst.
- [ ] Zielname und Ziel-IP-Adresse wurden erfasst.
- [ ] der benötigte Zielport wurde bestätigt.
- [ ] TCP und UDP wurden unterschieden.
- [ ] die Namensauflösung wurde geprüft.
- [ ] Ping wurde nicht mit einem Diensttest gleichgesetzt.
- [ ] der Zielport wurde vom betroffenen Client getestet.
- [ ] Erfolg, Ablehnung, Reset oder Zeitüberschreitung wurden unterschieden.
- [ ] ein funktionierender Vergleichsclient wurde geprüft.
- [ ] IPv4 und IPv6 wurden getrennt betrachtet.
- [ ] ein Listener auf dem Server wurde nachgewiesen.
- [ ] der zugehörige Prozess wurde bestimmt.
- [ ] die Bind-Adresse wurde kontrolliert.
- [ ] der Dienststatus wurde geprüft.
- [ ] Dienstprotokolle wurden ausgewertet.
- [ ] der Dienst wurde lokal über Loopback getestet.
- [ ] der Dienst wurde lokal über die Server-IP getestet.
- [ ] ein protokollspezifischer Test wurde durchgeführt.
- [ ] die lokale Server-Firewall wurde geprüft.
- [ ] die Client-Firewall wurde berücksichtigt.
- [ ] Netzwerk-Firewalls und ACLs wurden geprüft.
- [ ] Firewallregeln wurden auf Quelle, Ziel, Port und Protokoll geprüft.
- [ ] Hin- und Rückroute wurden berücksichtigt.
- [ ] asymmetrisches Routing wurde bei Bedarf geprüft.

- [ ] NAT und Portweiterleitungen wurden berücksichtigt.
- [ ] VPN- und Segmentierungsregeln wurden geprüft.
- [ ] Container- oder VM-Portzuordnungen wurden berücksichtigt.
- [ ] Reverse Proxy oder Load Balancer wurde geprüft.
- [ ] die Verbindung zum Backend wurde getestet.
- [ ] TLS und Zertifikate wurden berücksichtigt.
- [ ] Hostname, SNI und virtuelle Hosts wurden berücksichtigt.
- [ ] Anwendung, Authentifizierung und Berechtigung wurden abgegrenzt.
- [ ] MTU und Paketverlust wurden bei passenden Symptomen berücksichtigt.
- [ ] anwendungsspezifische Proxys wurden geprüft.
- [ ] bei sporadischen Fehlern wurden zeitgestempelte Tests durchgeführt.
- [ ] eine Paketaufzeichnung erfolgte nur mit Berechtigung.
- [ ] der Ausgangszustand wurde vor Änderungen dokumentiert.
- [ ] es wurde nur eine kontrollierbare Änderung durchgeführt.
- [ ] der Porttest wurde nach der Änderung wiederholt.
- [ ] das Anwendungsprotokoll wurde erneut getestet.
- [ ] die ursprüngliche Anwendung wurde erneut geprüft.
- [ ] unzulässige Quellnetze bleiben weiterhin gesperrt.
- [ ] temporäre Testfreigaben wurden entfernt.
- [ ] Ursache, Maßnahme und Nachweis wurden dokumentiert.

---

40. Schnellreferenz

Fehlerbild	wahrscheinlicher Untersuchungsbereich
Ping funktioniert, TCP-Port nicht	Listener, Firewall, Routing, NAT oder Rückweg
TCP-Verbindung wird sofort abgelehnt	kein Listener, falscher Port oder aktive Ablehnung
TCP-Verbindung läuft in Zeitüberschreitung	Firewall-Drop, Routing, NAT, Rückweg oder Ziel
Port funktioniert lokal, entfernt nicht	Bind-Adresse, Firewall, ACL oder Netzwerkpfad
Loopback funktioniert, Server-IP lokal nicht	Bind-Adresse oder lokale Firewall
Port ist offen, Anwendung antwortet nicht	Dienst, Protokoll, Backend oder Überlastung
TCP funktioniert, TLS scheitert	Zertifikat, SNI, Protokoll oder Cipher
HTTPS liefert `401`	Authentifizierung
HTTPS liefert `403`	Berechtigung oder Richtlinie
HTTPS liefert `502`	Proxy erreicht Backend nicht korrekt

Fehlerbild	wahrscheinlicher Untersuchungsbereich
HTTPS liefert `503`	Dienst oder Backend nicht verfügbar
HTTPS liefert `504`	Zeitüberschreitung zwischen Proxy und Backend
nur ein Client betroffen	Client-Firewall, Proxy, Route oder lokale Anwendung
nur ein Netz betroffen	ACL, Firewall, Routing oder Quellnetzregel
nur VPN-Clients betroffen	VPN-Route, VPN-Firewall, MTU oder DNS
intern erreichbar, extern nicht	NAT, Portweiterleitung, externe Firewall oder Provider
extern erreichbar, intern nicht	Hairpin-NAT, Split-DNS oder interne Firewall
IP funktioniert, Hostname nicht	DNS, SNI, virtueller Host oder Zertifikat
IPv4 funktioniert, IPv6 nicht	IPv6-Listener, Firewall, Route oder AAAA-Record
kleine Anfragen funktionieren, große nicht	MTU, Fragmentierung, Paketverlust oder Timeout
Verbindung scheitert sporadisch	Last, Porterschöpfung, defektes Backend oder Paketverlust
UDP-Test bleibt ohne Antwort	Dienst, Firewall, falsche Anfrage oder normales Protokollverhalten
SYN ohne Antwort	Drop, Routing, Ziel oder Rückweg
SYN gefolgt von RST	Port geschlossen oder aktive Ablehnung
Handshake vollständig, danach RST	Anwendung, Proxy, Firewall oder Protokoll
Handshake vollständig, keine Antwort	Dienst hängt, Backend wartet oder Überlastung

---

Merksatz

> Ein erreichbarer Host ist nicht automatisch ein erreichbarer Dienst. Geprüft werden müssen die vollständige Kommunikationsbeziehung aus Quelle, Zieladresse, Transportprotokoll und Zielport sowie anschließend Listener, Firewall, Rückweg und Anwendung. Ein erfolgreicher Porttest beweist den TCP-Verbindungsaufbau, aber noch nicht die Funktionsfähigkeit des Anwendungsprotokolls.

Quellen und weiterführende Dokumentation

- Microsoft Learn – Guidance for troubleshooting TCP/IP communication
- Microsoft Learn – Troubleshoot TCP/IP connectivity
- Microsoft Learn – Test-NetConnection
- Microsoft Learn – Test-Connection
- Microsoft Learn – Get-NetTCPConnection

- Microsoft Learn – Get-NetUDPEndpoint
- Microsoft Learn – Windows Defender Firewall with Advanced Security administration
- RFC 9293 – Transmission Control Protocol
- RFC 768 – User Datagram Protocol
- RFC 792 – Internet Control Message Protocol
- RFC 8201 – Path MTU Discovery for IPv6
- Wireshark – Display Filter Reference: TCP
- Wireshark – Display Filter Reference: UDP
- Wireshark User’s Guide – Building display filter expressions