# 3.12 Proxy- und Reverse-Proxy-Fehler analysieren

Ein Proxy befindet sich zwischen Client und Zielsystem. Je nach Proxyart verarbeitet, filtert, protokolliert oder vermittelt er Verbindungen. Dadurch kann eine direkte Verbindung funktionieren, während der Zugriff über den Proxy fehlschlägt – oder umgekehrt.

Die zentralen Fragen dieser Seite lauten:

> Welcher Proxy wird tatsächlich verwendet, wie wurde er ausgewählt und an welcher Stelle zwischen Client, Proxy und Zielsystem scheitert die Verbindung?

---

## 1. Sicherheits- und Aktionskennzeichnungen

| Kennzeichnung | Bedeutung |
|---|---|
| [RO] | Rein lesender Befehl, der normalerweise keine Konfiguration verändert |
| [TEST] | Aktiver Test, der Netzwerkverkehr erzeugt |
| [PRIV] | Administrator- oder Root-Rechte können erforderlich sein |
| [FILE] | Der Befehl schreibt Ausgaben in eine Datei |
| [SENS] | Die Ausgabe kann sensible Informationen enthalten |
| [CHANGE] | Der Befehl verändert eine Konfiguration |
| [DISRUPT] | Der Befehl kann Verbindungen oder Dienste beeinträchtigen |

Proxykonfigurationen und ausführliche HTTP-Ausgaben können interne Hostnamen, Benutzernamen, Cookies, Autorisierungsheader und Sitzungstoken enthalten. Sie dürfen nicht ungeprüft dokumentiert oder weitergegeben werden.

---

## 2. Proxyarten unterscheiden

| Proxyart | Position und Aufgabe |
|---|---|

| | |
|---|---|
| Forward Proxy | Vermittelt Verbindungen von Clients zu externen oder internen Zielen |
| Explicit Proxy | Client kennt Proxyadresse und Proxyport ausdrücklich |
| Transparent Proxy | Verkehr wird ohne ausdrückliche Proxykonfiguration umgeleitet |
| Reverse Proxy | Nimmt Anfragen für Serverdienste entgegen und leitet sie an Backends weiter |
| SOCKS-Proxy | Vermittelt TCP-Verbindungen und abhängig von Version und Werkzeug weitere Funktionen |
| Caching Proxy | Speichert Antworten zwischen, um Zugriffe zu beschleunigen oder Datenverkehr zu reduzieren |
| Filtering Proxy | Filtert Ziele, Kategorien, Inhalte oder Dateitypen |
| TLS-Inspection-Proxy | Entschlüsselt und untersucht TLS-Verbindungen entsprechend der Sicherheitsrichtlinie |
| PAC-Proxy | Proxy wird durch eine Proxy-Autokonfigurationsdatei ausgewählt |
| WPAD | Verfahren zur automatischen Ermittlung einer Proxykonfiguration |
| Application Proxy | Gilt nur für eine bestimmte Anwendung oder einen bestimmten Dienst |
| System Proxy | Wird durch Betriebssystemkomponenten oder mehrere Anwendungen verwendet |

**Wichtig:** Nicht jede Anwendung verwendet automatisch die systemweite Proxykonfiguration.

---

## 3. Forward Proxy und Reverse Proxy auseinanderhalten

**Forward Proxy:**

```
Client
  ↓
Forward Proxy
  ↓
Zielserver im Internet oder Netzwerk
```

Der Client oder das Betriebssystem kennt den Proxy normalerweise.

**Reverse Proxy:**

```
Client
  ↓
Reverse Proxy
  ↓
Backend-Anwendung
```

Der Client kennt normalerweise nur den öffentlichen Dienstnamen. Dass dahinter ein Reverse Proxy und ein oder mehrere Backends stehen, muss für ihn nicht sichtbar sein.

| Fehlerstelle | Typische Prüfung |
|---|---|
| Client zum Forward Proxy | DNS, Route, Port, Authentifizierung |
| Forward Proxy zum Zielserver | Proxyprotokoll, Filterregel, DNS und Ausgangsverbindung |
| Client zum Reverse Proxy | DNS, Port, TLS und Firewall |
| Reverse Proxy zum Backend | Backendadresse, Port, Protokoll, TLS und Anwendung |
| Backend selbst | Dienststatus, Listener, Protokolle und Antwortzeit |

---

## 4. Typische Fehlermeldungen einordnen

| Meldung oder Status | Typische Bedeutung |
|---|---|
| `Could not resolve proxy` | Proxyhostname konnte nicht aufgelöst werden |
| `Failed to connect to proxy` | Proxyadresse wurde ermittelt, Verbindung zum Proxyport schlägt fehl |
| `407 Proxy Authentication Required` | Forward Proxy verlangt eine Authentifizierung |
| `403 Forbidden` | Zugriff wird durch Zielserver, Proxy oder Richtlinie verweigert |
| `502 Bad Gateway` | Reverse Proxy erhielt keine gültige Antwort vom Backend |
| `503 Service Unavailable` | Dienst oder Backend ist nicht verfügbar |
| `504 Gateway Timeout` | Proxy erhielt innerhalb der vorgesehenen Zeit keine Backendantwort |
| `Connection refused` | Ziel erreichbar, aber Port wird nicht angenommen |
| `Connection timed out` | Keine rechtzeitige Antwort; Filterung, Routing oder Ausfall möglich |

| Meldung oder Status | Typische Bedeutung |
| --- | --- |
| `ERR_PROXY_CONNECTION_FAILED` | Browser konnte den konfigurierten Proxy nicht verwenden |
| `ERR_TUNNEL_CONNECTION_FAILED` | Aufbau eines Proxytunnels, häufig per CONNECT, ist fehlgeschlagen |
| Zertifikat nicht vertrauenswürdig | Zertifikatskette, TLS-Inspection oder falsches Zertifikat prüfen |
| Umleitungsschleife | Proxy-, Anwendungsschema- oder Hostheaderkonfiguration prüfen |
| Nur Browser funktioniert | Anwendung verwendet möglicherweise andere Proxyeinstellungen |
| Browser funktioniert nicht, CLI schon | Browser-, PAC-, Zertifikats- oder Erweiterungskonfiguration prüfen |

Ein HTTP-Statuscode beweist noch nicht eindeutig, welche Komponente ihn erzeugt hat. Antwortheader und Protokolle müssen miteinander verglichen werden.

---

## 5. Den betroffenen Datenfluss dokumentieren

Vor der Diagnose müssen folgende Angaben bekannt sein:

```
Client:
Client-IP:
Benutzerkontext:
Anwendung:
Ziel-URL:
Zielhostname:
Zielport:
Proxyart:
Proxyhostname:
Proxy-IP:
Proxyport:
PAC-URL:
Verwendete Authentifizierung:
Zeitpunkt:
Fehlermeldung:
```

Zusätzlich prüfen:

- Tritt der Fehler bei allen Benutzern auf?
- Tritt er bei allen Anwendungen auf?
- Sind nur interne oder externe Ziele betroffen?
- Funktioniert HTTP, aber HTTPS nicht?
- Funktioniert der Zugriff ohne VPN?
- Tritt der Fehler nur in einem bestimmten Netzwerk auf?
- Ist nur ein einzelner Hostname oder eine gesamte Kategorie betroffen?
- Funktioniert derselbe Zugriff auf einem Vergleichssystem?

---

## 6. Mehrere Proxykonfigurationen berücksichtigen

Auf demselben System können unterschiedliche Einstellungen gleichzeitig existieren:

- Betriebssystem-Proxy,
- WinHTTP-Proxy,
- benutzerbezogene Windows-Einstellungen,
- Browserkonfiguration,
- PAC-Datei,
- WPAD,
- Umgebungsvariablen,
- Anwendungskonfiguration,
- Java-Proxyparameter,
- Paketmanagerkonfiguration,
- Container-Umgebungsvariablen,
- Gruppenrichtlinie oder MDM-Profil,
- VPN- oder Sicherheitssoftware,
- transparenter Netzwerkproxy.

**Typisches Fehlerbild:**

```
Browser verwendet PAC-Datei
PowerShell-Dienst verwendet WinHTTP
CLI-Programm verwendet HTTPS_PROXY
Anwendung besitzt eigene Proxykonfiguration
```

Deshalb muss immer genau geprüft werden, **welche Anwendung unter welchem Benutzerkonto** betroffen ist.

---

## 7. Proxy-Umgebungsvariablen prüfen

Häufig verwendete Variablen:

```
http_proxy
https_proxy
all_proxy
no_proxy
HTTP_PROXY
HTTPS_PROXY
ALL_PROXY
NO_PROXY
```

Die unterstützte Schreibweise hängt von Anwendung und Bibliothek ab. Bei curl besitzt insbesondere `http_proxy` eine besondere Behandlung und wird aus Sicherheitsgründen nur in Kleinbuchstaben ausgewertet.

**Windows PowerShell:**

```
[RO][SENS] Get-ChildItem Env: |
    Where-Object Name -Match '^(http|https|all|no)_proxy$' |
    Sort-Object Name
```

**Windows CMD:**

```
[RO][SENS] set | findstr /I "http_proxy https_proxy all_proxy no_proxy"
```

**Linux und macOS:**

```
[RO][SENS] env | grep -iE '^(http|https|all|no)_proxy='
```

**Nur Variablennamen ohne Werte anzeigen:**

```
[RO] env | sed -n 's/^\([^=]*[Pp][Rr][Oo][Xx][Yy]\)=.*/\1/p'
```

Proxy-URLs können Zugangsdaten enthalten:

```
http://benutzer:passwort@proxy.example.internal:8080
```

Solche Werte dürfen nicht in Tickets, Bildschirmfotos oder Dokumentationen übernommen werden.

---

## 8. Windows: WinHTTP-Proxy prüfen

WinHTTP wird von bestimmten Windows-Diensten und Anwendungen verwendet. Seine Konfiguration kann von den benutzerbezogenen Proxy-Einstellungen abweichen.

**Grundlegende WinHTTP-Konfiguration:**

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

**Erweiterte WinHTTP-Konfiguration:**

```
[RO][SENS] netsh winhttp show advproxy
```

Mögliche Informationen:

- Proxyserver,
- Proxyport,
- Umgehungsliste,
- PAC-URL,
- automatische Erkennung,
- direkte Verbindung.

**Wichtig:**

```
Direkter Zugriff im Browser funktioniert
≠
WinHTTP-Dienst funktioniert
```

und:

```
WinHTTP funktioniert
≠
Browser verwendet dieselbe Konfiguration
```

Änderungsbefehle wie `netsh winhttp reset proxy`, `set proxy` oder `import proxy` dürfen nicht als rein lesende Diagnosebefehle verwendet werden.

---

## 9. Windows: benutzerbezogene Proxyeinstellungen prüfen

**Relevante Benutzereinstellungen auslesen:**

```
[RO][SENS] Get-ItemProperty `
    "HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings" |
    Select-Object ProxyEnable, ProxyServer, ProxyOverride, AutoConfigURL, AutoDetect
```

Je nach Windows-Version und Richtlinienquelle können einzelne Eigenschaften fehlen oder anders verwaltet werden.

| Eigenschaft | Typische Bedeutung |
|---|---|
| `ProxyEnable` | Manuell eingetragener Proxy aktiviert |
| `ProxyServer` | Proxyadresse beziehungsweise protokollspezifische Zuordnungen |
| `ProxyOverride` | Umgehungsliste |
| `AutoConfigURL` | URL einer PAC-Datei |
| `AutoDetect` | Automatische Erkennung aktiviert |

**Nur vorhandene Werte einzeln prüfen:**

```
[RO][SENS] Get-ItemProperty `
    "HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings" `
    -ErrorAction SilentlyContinue
```

Zu kontrollieren sind:

- Wird ein manueller Proxy verwendet?
- Ist zusätzlich eine PAC-Datei eingetragen?
- Ist automatische Erkennung aktiviert?
- Enthält die Umgehungsliste das Ziel?
- Wird die Einstellung durch Gruppenrichtlinien verwaltet?
- Tritt der Fehler nur bei einem Benutzerprofil auf?

Registrywerte sollten nicht direkt verändert werden, solange Verwaltungsquelle und Sollzustand nicht geklärt sind.

---

## 10. Windows: Richtlinien und Benutzerkontext prüfen

**Angewendete Gruppenrichtlinien anzeigen:**

```
[RO][FILE][SENS] gpresult /h "%TEMP%\gpresult-proxy.html"
```

Die erzeugte HTML-Datei befindet sich im temporären Verzeichnis des aktuellen Benutzers.

**Nur Textausgabe:**

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

Zu prüfen sind:

- Computer- oder Benutzerrichtlinie,
- Proxyserver,
- PAC-URL,
- Browserrichtlinien,
- Zertifikatsverteilung,
- Sicherheitszonen,
- lokale Umgehungsmöglichkeiten,
- Richtlinien des Sicherheitsprodukts.

**Benutzerkontext bestimmen:**

```
[RO] whoami
```

Ein Dienstkonto, Systemkonto oder anderer Benutzer kann eine andere Proxy- und Zertifikatskonfiguration besitzen als der interaktiv angemeldete Benutzer.

---

## 11. macOS-Proxyeinstellungen prüfen

Unter macOS werden Proxyeinstellungen pro Netzwerkdienst verwaltet, beispielsweise getrennt für WLAN und Ethernet.

**Netzwerkdienste anzeigen:**

```
[RO] networksetup -listallnetworkservices
```

Ein Sternchen vor einem Dienstnamen weist in der Ausgabe darauf hin, dass der Netzwerkdienst deaktiviert ist.

**HTTP-Proxy eines Netzwerkdienstes:**

```
[RO][SENS] networksetup -getwebproxy "Wi-Fi"
```

**HTTPS-Proxy:**

```
[RO][SENS] networksetup -getsecurewebproxy "Wi-Fi"
```

**SOCKS-Proxy:**

```
[RO][SENS] networksetup -getsocksfirewallproxy "Wi-Fi"
```

**PAC-URL:**

```
[RO][SENS] networksetup -getautoproxyurl "Wi-Fi"
```

**Proxy-Umgehungsliste:**

```
[RO][SENS] networksetup -getproxybypassdomains "Wi-Fi"
```

`Wi-Fi` muss durch den tatsächlich verwendeten Netzwerkdienst ersetzt werden.

**Alle aktiven Proxyinformationen zusammengefasst anzeigen:**

```
[RO][SENS] scutil --proxy
```

Zu kontrollieren sind:

- Welcher Netzwerkdienst ist aktiv?
- Ist der Proxy für diesen Dienst aktiviert?
- Stimmen Proxyhostname und Port?

- Ist eine PAC-URL aktiv?
- Existieren Ausnahmen?
- Wird die Konfiguration durch ein MDM-Profil verwaltet?
- Ändert ein VPN-Client die Proxykonfiguration?

---

## 12. Linux-Proxyeinstellungen prüfen

Linux besitzt keine für alle Distributionen und Anwendungen einheitliche Proxyverwaltung. Zu prüfen sind insbesondere:

- Umgebungsvariablen,
- Desktopumgebung,
- Paketmanager,
- systemd-Dienstkonfiguration,
- Anwendungskonfiguration,
- Containerkonfiguration,
- Shell-Startdateien,
- zentrale Konfigurationsverwaltung.

**Umgebungsvariablen:**

```
[RO][SENS] env | grep -iE '^(http|https|all|no)_proxy='
```

**Systemweite Umgebungsdatei prüfen, sofern vorhanden:**

```
[RO][SENS] test -f /etc/environment && sed -n '/[Pp][Rr][Oo][Xx][Yy]/p' /etc/environment
```

**systemd-Umgebung eines Dienstes anzeigen:**

```
[RO][PRIV][SENS] sudo systemctl show DIENSTNAME \
    --property=Environment \
    --property=EnvironmentFiles
```

**Vollständige Unit-Konfiguration anzeigen:**

```
[RO][PRIV][SENS] sudo systemctl cat DIENSTNAME
```

**Laufenden Prozess und Benutzer prüfen:**

```
[RO] systemctl status DIENSTNAME
```

`DIENSTNAME` muss durch den tatsächlichen Dienst ersetzt werden.

Ein interaktiver Shell-Test beweist nicht, dass ein systemd-Dienst dieselben Umgebungsvariablen verwendet.

---

## 13. Manuelle Proxyverbindung mit curl testen

**HTTP-Ziel ausdrücklich über einen Proxy testen:**

```
[TEST][SENS] curl -v \
  --proxy http://proxy.example.internal:8080 \
  http://example.com/
```

**HTTPS-Ziel über einen HTTP-Proxy testen:**

```
[TEST][SENS] curl -v \
  --proxy http://proxy.example.internal:8080 \
  https://example.com/
```

Bei HTTPS verwendet curl über einen HTTP-Proxy normalerweise einen CONNECT-Tunnel zum Ziel.

**Nur Antwortheader abrufen:**

```
[TEST][SENS] curl -I \
  --proxy http://proxy.example.internal:8080 \
  https://example.com/
```

**Zeitlimit setzen:**

```
[TEST][SENS] curl -v \
  --connect-timeout 5 \
  --max-time 15 \
  --proxy http://proxy.example.internal:8080 \
```

```
    https://example.com/
```

**Wichtige Ausgaben:**

```
Trying PROXY_IP:PORT
Connected to proxy
CONNECT ziel.example:443
HTTP/1.1 200 Connection established
Proxy-Authenticate
HTTP-Statuscode
TLS-Handshake
Zertifikatskette
```

`curl -v` kann sensible Header ausgeben. Vor der Dokumentation müssen Zugangsdaten, Cookies und Token entfernt werden.

---

## 14. Direkten Zugriff und Proxyzugriff vergleichen

**Proxy ausdrücklich umgehen:**

```
[TEST][SENS] curl -v --noproxy "*" https://example.com/
```

**Proxy ausdrücklich verwenden:**

```
[TEST][SENS] curl -v \
    --proxy http://proxy.example.internal:8080 \
    https://example.com/
```

**Nur für ein bestimmtes Ziel den Proxy umgehen:**

```
[TEST][SENS] curl -v \
    --noproxy "example.com" \
    https://example.com/
```

**Vergleich:**

| Direkter Zugriff | Proxyzugriff | Mögliche Eingrenzung |
|---|---|---|
| Erfolgreich | Fehlerhaft | Proxy, Authentifizierung, PAC oder Filterrichtlinie prüfen |
| Fehlerhaft | Erfolgreich | Direkter Weg blockiert oder Proxy ist vorgeschrieben |
| Beide fehlerhaft | Ziel, DNS, Clientnetz oder allgemeine Störung prüfen | |
| Beide erfolgreich | Fehler ist möglicherweise anwendungsspezifisch | |
| Nur Browser fehlerhaft | Browserprofil, PAC, Zertifikat oder Erweiterung prüfen | |
| Nur Systemdienst fehlerhaft | WinHTTP-, Dienstkonto- oder Dienstumgebung prüfen | |

Ein direkter Test darf nur durchgeführt werden, wenn die Sicherheitsrichtlinie einen Proxy-Bypass erlaubt.

---

## 15. Proxyhostname, Port und Erreichbarkeit prüfen

**Namensauflösung des Proxys:**

Windows:

```
[TEST] Resolve-DnsName proxy.example.internal
```

Linux:

```
[TEST] getent ahosts proxy.example.internal
```

macOS:

```
[TEST] dscacheutil -q host -a name proxy.example.internal
```

**Proxyport testen:**

Windows:

```
[TEST] Test-NetConnection proxy.example.internal -Port 8080 -InformationLevel Detailed
```

Linux und macOS:

```
[TEST] nc -vz -w 5 proxy.example.internal 8080
```

**Route zum Proxy prüfen:**

Windows:

```
[RO] Find-NetRoute -RemoteIPAddress PROXY_IP
```

Linux:

```
[RO] ip route get PROXY_IP
```

macOS:

```
[RO] route -n get PROXY_IP
```

Ein erfolgreicher TCP-Porttest beweist nur, dass eine Verbindung zum Proxyport aufgebaut werden konnte. Er beweist nicht, dass der Proxy die gewünschte Zielverbindung erlaubt.

---

## 16. PAC-Datei und automatische Proxyauswahl prüfen

Eine PAC-Datei enthält JavaScript und liefert über die Funktion `FindProxyForURL()` eine Proxyentscheidung.

Typische Rückgabewerte:

```
DIRECT
PROXY proxy.example.internal:8080
HTTPS proxy.example.internal:8443
SOCKS socks.example.internal:1080
```

Mehrere Möglichkeiten können in einer Reihenfolge angegeben werden:

```
PROXY proxy1.example.internal:8080;
PROXY proxy2.example.internal:8080;
DIRECT
```

**Konfigurierte PAC-URL ermitteln:**

- Windows: `AutoConfigURL` beziehungsweise WinHTTP-Advanced-Proxy prüfen
- macOS: `networksetup -getautoproxyurl` oder `scutil --proxy`
- Linux: Desktop- oder Anwendungskonfiguration prüfen

**PAC-Datei kontrolliert abrufen:**

```
[TEST][SENS] curl -v "http://config.example.internal/proxy.pac"
```

**Ohne möglicherweise bereits gesetzte Proxyvariablen abrufen:**

```
[TEST][SENS] curl -v --noproxy "*" \
  "http://config.example.internal/proxy.pac"
```

Zu prüfen sind:

- Ist die PAC-URL per DNS erreichbar?
- Ist der Webserver erreichbar?
- Wird ein erfolgreicher HTTP-Status geliefert?
- Ist der MIME-Typ plausibel?
- Ist die Datei syntaktisch gültig?
- Welche Entscheidung trifft sie für die konkrete URL?
- Werden interne und externe Namen unterschiedlich behandelt?
- Gibt es Zeitverzögerungen durch DNS-Abfragen?
- Enthält die Datei veraltete Proxynamen?
- Wird die PAC-Datei zwischengespeichert?

PAC-Dateien sind ausführbarer JavaScript-Code und dürfen nur aus vertrauenswürdigen Quellen bezogen werden.

---

## 17. WPAD und automatische Erkennung beurteilen

WPAD kann eine Proxykonfiguration automatisch bereitstellen. Abhängig von der Umgebung kann die Ermittlung unter anderem über DHCP oder DNS erfolgen.

Mögliche Fehler:

- falscher WPAD-DNS-Eintrag,
- WPAD-Server nicht erreichbar,
- veraltete PAC-Datei,
- unterschiedliche Suchdomänen,
- VPN verändert DNS oder Suchsuffixe,
- Client verwendet automatische Erkennung nicht,
- Sicherheitsrichtlinie blockiert WPAD,
- nicht autorisierter WPAD-Server.

**Windows-DNS-Suffixe prüfen:**

```
[RO] Get-DnsClient |
    Select-Object InterfaceAlias, ConnectionSpecificSuffix
```

**Windows-DNS-Konfiguration:**

```
[RO] Get-DnsClientGlobalSetting
```

**Linux-Suchdomänen prüfen:**

```
[RO] resolvectl status
```

Falls `resolvectl` nicht vorhanden ist:

```
[RO] cat /etc/resolv.conf
```

**macOS-DNS-Konfiguration:**

```
[RO][SENS] scutil --dns
```

WPAD darf nicht durch willkürlich angelegte DNS-Einträge getestet werden. Eine fehlerhafte oder manipulierte WPAD-Konfiguration kann den gesamten Webverkehr umleiten.

---

## 18. Proxy-Umgehungslisten prüfen

Eine Umgehungsliste bestimmt, welche Ziele direkt und nicht über den Proxy erreicht werden.

Typische Einträge:

```
localhost
127.0.0.1
::1
*.example.internal
192.168.0.0/16
<local>
```

Die unterstützte Syntax unterscheidet sich zwischen Betriebssystemen und Anwendungen. Nicht jede Anwendung versteht CIDR-Netze, Platzhalter oder `<local>` gleich.

**Typische Fehler:**

- Hostname passt nicht zum Platzhaltermuster.
- Kurzname wird umgangen, FQDN jedoch nicht.
- IP-Adresse wird umgangen, Hostname dagegen nicht.
- Falscher Trenner wurde verwendet.
- Portnummer wurde unnötig in das Muster aufgenommen.
- IPv6-Ziel fehlt.
- `NO_PROXY` enthält Leerzeichen oder eine ungeeignete Syntax.
- Anwendung interpretiert führende Punkte anders.
- Interner Dienst wird versehentlich über externen Proxy gesendet.

**curl-Entscheidung sichtbar machen:**

```
[TEST][SENS] curl -v https://service.example.internal/
```

In der ausführlichen Ausgabe kann curl anzeigen, welche Proxy- beziehungsweise `NO_PROXY`-Variable berücksichtigt wurde.

---

## 19. Proxy-Authentifizierung untersuchen

Ein Forward Proxy kann unter anderem folgende Verfahren verwenden:

- Basic,
- Digest,
- NTLM,

- Negotiate beziehungsweise Kerberos,
- zertifikatsbasierte Authentifizierung,
- IP- oder gerätebasierte Freigabe.

**Proxyantwort ohne Zugangsdaten prüfen:**

```
[TEST][SENS] curl -v \
  --proxy http://proxy.example.internal:8080 \
  https://example.com/
```

Auf folgende Header achten:

```
HTTP/1.1 407 Proxy Authentication Required
Proxy-Authenticate: Basic
Proxy-Authenticate: NTLM
Proxy-Authenticate: Negotiate
```

**Sicherheitsregeln:**

- Passwörter nicht direkt in die Befehlszeile schreiben.
- Zugangsdaten nicht in Shell-History speichern.
- Autorisierungsheader nicht in Tickets kopieren.
- Keine produktiven Kennwörter in PAC-Dateien oder Umgebungsvariablen ablegen.
- Authentifizierungsverfahren nicht ohne Freigabe herabsetzen.
- Uhrzeit und Domänenanmeldung bei Kerberos-Problemen prüfen.

**Benutzerkontext prüfen:**

Windows:

```
[RO] whoami
```

Linux und macOS:

```
[RO] id
```

Ein Hintergrunddienst besitzt möglicherweise keine interaktive Benutzersitzung und kann daher keine integrierte Proxyauthentifizierung durchführen.

## 20. TLS-Inspection und Zertifikatsfehler analysieren

Bei TLS-Inspection baut die Sicherheitskomponente getrennte TLS-Verbindungen auf:

```
Client
  ↓ TLS-Verbindung 1
Inspection-Proxy
  ↓ TLS-Verbindung 2
Zielserver
```

Der Client sieht dabei ein vom Inspection-System ausgestelltes Zertifikat.

**Zertifikat über den vorgesehenen Verbindungsweg prüfen:**

```
[TEST][SENS] curl -v https://example.com/
```

**TLS-Verbindung zu einem HTTPS-Ziel über HTTP-Proxy untersuchen:**

```
[TEST][SENS] curl -v \
   --proxy http://proxy.example.internal:8080 \
   https://example.com/
```

Zu prüfen sind:

- Aussteller des präsentierten Zertifikats,
- Hostname im Zertifikat,
- Gültigkeitszeitraum,
- Vertrauenskette,
- Unternehmens-Stammzertifikat,
- Zertifikatsspeicher der betroffenen Anwendung,
- Systemzeit,
- TLS-Version,
- SNI,
- Ausnahmen von der TLS-Inspection.

**Typisches Fehlerbild:**

```
Browser funktioniert
CLI-Anwendung meldet unbekannte Zertifizierungsstelle
```

Mögliche Ursache:

- Browser und Anwendung verwenden unterschiedliche Zertifikatsspeicher.

Die Zertifikatsprüfung darf nicht dauerhaft deaktiviert werden. Optionen wie `curl -k` dienen höchstens einer genehmigten Eingrenzung und stellen keine Fehlerbehebung dar.

---

## 21. Reverse Proxy vom Backend trennen

Ein Reverse-Proxy-Datenfluss besteht mindestens aus zwei Verbindungen:

```
Verbindung 1:
Client → Reverse Proxy

Verbindung 2:
Reverse Proxy → Backend
```

Beide Verbindungen können unterschiedliche Eigenschaften besitzen:

| Eigenschaft | Clientseite | Backendseite |
|---|---|---|
| Protokoll | HTTPS | HTTP oder HTTPS |
| Port | 443 | beispielsweise 80, 8080 oder 8443 |
| DNS | öffentlicher Dienstname | interner Backendname |
| Zertifikat | öffentliches oder internes Frontendzertifikat | gegebenenfalls internes Backendzertifikat |
| Authentifizierung | Benutzeranmeldung | Dienst- oder keine zusätzliche Anmeldung |
| Firewall | Client zum Proxy | Proxy zum Backend |
| Timeout | Frontendtimeout | Upstreamtimeout |

Ein erfolgreicher Zugriff auf den Reverse-Proxy-Port beweist nicht, dass das Backend erreichbar ist.

---

## 22. Reverse Proxy systematisch prüfen

**Schritt 1 – DNS des öffentlichen Dienstnamens:**

Windows:

```
[TEST] Resolve-DnsName service.example.net
```

Linux und macOS:

```
[TEST] dig service.example.net A
[TEST] dig service.example.net AAAA
```

**Schritt 2 – Frontend-Port:**

Windows:

```
[TEST] Test-NetConnection service.example.net -Port 443
```

Linux und macOS:

```
[TEST] nc -vz -w 5 service.example.net 443
```

**Schritt 3 – Frontend-Anwendung:**

```
[TEST][SENS] curl -v https://service.example.net/
```

**Schritt 4 – Backend vom Reverse-Proxy-System aus:**

```
[TEST][SENS] curl -v http://backend.example.internal:8080/
```

**Schritt 5 – Backend-Port:**

```
[TEST] nc -vz -w 5 backend.example.internal 8080
```

**Schritt 6 – Protokolle des Reverse Proxys:**

- Zugriffsprotokoll,
- Fehlerprotokoll,
- Upstreamstatus,
- Antwortzeit,

- ausgewähltes Backend,
- TLS-Fehler,
- Namensauflösungsfehler.

Der Backendtest muss vom Reverse-Proxy-System beziehungsweise aus dessen Netzwerk- oder Containerkontext erfolgen. Ein Test von einem Administrator-PC kann einen anderen Netzwerkpfad verwenden.

---

## 23. Wichtige Reverse-Proxy-Header prüfen

Reverse Proxys übermitteln dem Backend häufig Informationen über die ursprüngliche Anfrage.

Typische Header:

```
Host
X-Forwarded-For
X-Forwarded-Host
X-Forwarded-Proto
Forwarded
```

| Header | Typische Aufgabe |
|---|---|
| `Host` | Ursprünglich angesprochener Hostname |
| `X-Forwarded-For` | Ursprüngliche Clientadresse beziehungsweise Proxykette |
| `X-Forwarded-Host` | Ursprünglich verwendeter Host |
| `X-Forwarded-Proto` | Ursprüngliches Schema wie `http` oder `https` |
| `Forwarded` | Standardisierter Header für Proxyinformationen |

**Mögliche Fehler:**

- Backend erhält falschen Hostheader.
- Anwendung erkennt HTTPS nicht und erzeugt HTTP-Links.
- Endlose HTTP-zu-HTTPS-Umleitung.
- Client-IP wird nicht korrekt übergeben.
- Anwendung vertraut beliebigen `X-Forwarded-For`-Werten.
- Mehrere Proxys erzeugen eine fehlerhafte Headerkette.
- Anwendung ist nicht für vertrauenswürdige Proxys konfiguriert.

Forwarded-Header dürfen nur von ausdrücklich vertrauenswürdigen Proxys als verlässlich behandelt werden.

## 24. Reverse-Proxy-Statuscodes untersuchen

| Status | Mögliche Ursache |
|---|---|
| 400 | Ungültige Anfrage, fehlerhafter Hostheader oder Protokollfehler |
| 401 | Anwendung verlangt Authentifizierung |
| 403 | Zugriff durch Proxy oder Backend verweigert |
| 404 | Falscher Host, Pfad oder Backendrouting |
| 408 | Anfrage wurde nicht rechtzeitig vollständig empfangen |
| 413 | Anfrage oder Upload überschreitet eine Größenbegrenzung |
| 421 | Anfrage wurde einem unpassenden Ziel beziehungsweise virtuellen Host zugeordnet |
| 429 | Rate Limit wurde erreicht |
| 431 | Header sind zu groß |
| 499 | Nicht standardisierter, insbesondere bei NGINX verwendeter Status für Clientabbruch |
| 500 | Interner Anwendungs- oder Proxyfehler |
| 502 | Backend nicht erreichbar oder ungültige Backendantwort |
| 503 | Kein verfügbares Backend oder Dienst absichtlich nicht verfügbar |
| 504 | Backend antwortet nicht rechtzeitig |

Zur eindeutigen Zuordnung müssen Antwortheader, Proxyprotokoll und Backendprotokoll zum selben Zeitpunkt verglichen werden.

## 25. NGINX-Reverse-Proxy prüfen

Die folgenden Befehle gelten nur, wenn NGINX tatsächlich eingesetzt wird.

**Konfigurationssyntax prüfen:**

```
[RO][PRIV][SENS] sudo nginx -t
```

Dieser Befehl prüft die Konfiguration, lädt sie aber nicht neu.

**Vollständig aufgelöste Konfiguration anzeigen:**

```
[RO][PRIV][SENS] sudo nginx -T
```

Die Ausgabe kann Zertifikatspfade, interne Hostnamen und weitere sensible Konfigurationswerte enthalten.

**Dienststatus:**

```
[RO] systemctl status nginx
```

**Prozess und Listener:**

```
[RO] ss -ltnp
```

**Dienstprotokolle über systemd:**

```
[RO][PRIV][SENS] sudo journalctl -u nginx --since "-15 minutes"
```

Zu prüfen sind:

- passende `server_name`-Direktive,
- `listen`-Adresse und Port,
- richtige `location`,
- korrektes `proxy_pass`-Ziel,
- Backendprotokoll HTTP oder HTTPS,
- Namensauflösung des Backends,
- TLS-SNI zum Backend,
- Weitergabe benötigter Header,
- Verbindungs- und Antworttime-outs,
- Uploadgrößen,
- WebSocket- beziehungsweise Upgrade-Header,
- verfügbares Backend.

Ein erfolgreicher `nginx -t` beweist nur, dass NGINX die Konfiguration syntaktisch akzeptiert. Er beweist nicht, dass das Backend erreichbar ist.

## 26. Container und Reverse Proxy prüfen

Bei containerisierten Anwendungen entstehen zusätzliche Ebenen:

```
Client
  ↓
Hostport
  ↓
Reverse-Proxy-Container
  ↓
Docker-Netzwerk
  ↓
Backend-Container
  ↓
Anwendung
```

**Container und Ports anzeigen:**

```
[RO] docker ps --format 'table {{.Names}}\t{{.Ports}}'
```

**Netzwerke anzeigen:**

```
[RO] docker network ls
```

**Netzwerk untersuchen:**

```
[RO][SENS] docker network inspect NETZWERKNAME
```

**Containerprotokolle:**

```
[RO][SENS] docker logs --tail 100 CONTAINERNAME
```

**Umgebungsvariablen eines Containers kontrollieren:**

```
[RO][SENS] docker inspect CONTAINERNAME
```

Zu prüfen sind:

- Befinden sich Proxy und Backend im selben Docker-Netzwerk?
- Wird der richtige Containername als DNS-Name verwendet?
- Stimmt der interne Containerport?
- Wird versehentlich der veröffentlichte Hostport als Containerziel verwendet?
- Lauscht die Anwendung auf `0.0.0.0` beziehungsweise einer erreichbaren Containeradresse?
- Startet das Backend später als der Proxy?
- Ist der Backendname nach einer Umbenennung veraltet?
- Enthält der Container Proxy-Umgebungsvariablen?
- Umgeht `NO_PROXY` interne Containerziele korrekt?

---

## 27. NO_PROXY bei Containern und internen Diensten prüfen

Ein Container kann ausgehende Verbindungen über einen Forward Proxy senden. Interne Dienste sollten abhängig vom Sollzustand möglicherweise direkt erreichbar sein.

Beispiel:

```
HTTP_PROXY=http://proxy.example.internal:8080
HTTPS_PROXY=http://proxy.example.internal:8080
NO_PROXY=localhost,127.0.0.1,backend,backend.example.internal
```

**Typisches Fehlerbild:**

```
Anwendung möchte internen Backend-Container erreichen
↓
Backendname fehlt in NO_PROXY
↓
Anfrage wird an Unternehmensproxy gesendet
↓
Proxy kann internen Containernamen nicht auflösen
↓
Verbindung schlägt fehl
```

**Konfiguration anzeigen:**

```
[RO][SENS] docker inspect CONTAINERNAME
```

Die genaue `NO_PROXY`-Syntax hängt von Anwendung und verwendeter Bibliothek ab. Änderungen müssen deshalb mit deren Dokumentation abgeglichen werden.

---

## 28. DNS-Auflösung aus dem richtigen Kontext prüfen

Client, Forward Proxy, Reverse Proxy und Backend können unterschiedliche DNS-Server und Antworten verwenden.

| System | Muss auflösen können |
|---|---|
| Client | Proxyhostname und gegebenenfalls Zielhostname |
| Forward Proxy | Externes oder internes Ziel |
| Reverse Proxy | Backendhostname |
| Backend | Abhängige Datenbanken, APIs und Dienste |
| Container | Containername oder internen DNS-Namen |

**Windows:**

```
[TEST] Resolve-DnsName HOSTNAME
```

**Linux:**

```
[TEST] getent ahosts HOSTNAME
```

**macOS:**

```
[TEST] dscacheutil -q host -a name HOSTNAME
```

Der Befehl muss auf dem System ausgeführt werden, das die jeweilige Verbindung tatsächlich aufbaut.

**Typisches Beispiel:**

```
Administrator-PC kann backend.example.internal auflösen
Reverse-Proxy-Container kann den Namen nicht auflösen
```

Der erfolgreiche Test vom Administrator-PC ist dann für die Backendverbindung nicht ausreichend.

---

## 29. Zeitüberschreitungen systematisch unterscheiden

Ein Proxyweg kann mehrere Time-outs besitzen:

```
Client-Verbindungszeit
Proxy-Verbindungszeit zum Backend
Zeit zum Senden der Anfrage
Zeit bis zum ersten Antwortbyte
Zeit zwischen Antwortdaten
Gesamtdauer der Anfrage
```

**curl-Zeitmessung:**

```
[TEST][SENS] curl -sS -o /dev/null \
    -w 'DNS: %{time_namelookup}\nConnect: %{time_connect}\nTLS: %{time_appconnect}\nStartTransfer: %{time_starttransfer}\nTotal: %{time_total}\nHTTP: %{http_code}\n' \
    https://service.example.net/
```

**Über einen expliziten Proxy:**

```
[TEST][SENS] curl -sS -o /dev/null \
    --proxy http://proxy.example.internal:8080 \
    -w 'DNS: %{time_namelookup}\nConnect: %{time_connect}\nTLS: %{time_appconnect}\nStartTransfer: %{time_starttransfer}\nTotal: %{time_total}\nHTTP: %{http_code}\n' \
    https://service.example.net/
```

Die Zeitwerte müssen mit Proxy- und Backendprotokollen verglichen werden. Ein `504` beweist nicht automatisch, dass das Backend vollständig ausgefallen ist; es kann lediglich später als das konfigurierte Zeitlimit geantwortet haben.

---

## 30. Paketmitschnitt für Proxyfehler erstellen

### Verkehr zwischen Client und Forward Proxy:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i INTERFACE -nn \
    'host PROXY_IP and tcp port PROXY_PORT' \
    -w client-proxy.pcap
```

### Verkehr zwischen Reverse Proxy und Backend:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i INTERFACE -nn \
    'host BACKEND_IP and tcp port BACKEND_PORT' \
    -w proxy-backend.pcap
```

### Windows mit Dumpcap – Schnittstellen anzeigen:

```
[RO] dumpcap -D
```

### Windows – Proxyverkehr aufzeichnen:

```
[TEST][PRIV][FILE][SENS] dumpcap -i 1 `
    -f "host PROXY_IP and tcp port PROXY_PORT" `
    -w proxy-test.pcapng
```

### Wireshark-Anzeigefilter:

| Aufgabe | Filter |
|---|---|
| Proxyport 8080 | `tcp.port == 8080` |
| HTTP-Statuscodes | `http.response.code` |
| Status 407 | `http.response.code == 407` |
| Status 502 | `http.response.code == 502` |
| Status 504 | `http.response.code == 504` |
| HTTP-CONNECT | `http.request.method == "CONNECT"` |
| TLS-Handshake | `tls.handshake` |
| TCP-Reset | `tcp.flags.reset == 1` |
| TCP-Wiederholungen | `tcp.analysis.retransmission` |

Verschlüsselter Datenverkehr zeigt ohne zulässige Entschlüsselung nicht den vollständigen HTTP-Inhalt.

---

## 31. Typische Fehlerbilder systematisch eingrenzen

**Fall A – Proxyhostname kann nicht aufgelöst werden**

Prüfen:

- Proxyname,
- DNS-Server,
- DNS-Suffix,
- VPN,
- PAC-Datei,
- veralteter Proxyname.

---

**Fall B – Proxyport ist nicht erreichbar**

Prüfen:

- Route,
- Firewall,
- Proxybetrieb,
- falscher Port,
- nur aus bestimmten VLANs erlaubt,
- IPv4- oder IPv6-Auswahl.

---

**Fall C – HTTP 407**

Prüfen:

- Authentifizierungsverfahren,
- Benutzerkonto,
- Dienstkonto,
- Domänenverbindung,
- Systemzeit,
- Kerberos,
- gespeicherte Anmeldeinformationen,
- Unterstützung der Anwendung.

---

**Fall D – Browser funktioniert, Dienst nicht**

Prüfen:

- WinHTTP,
- Dienstkonto,
- Umgebungsvariablen,
- Zertifikatsspeicher,
- PAC-Unterstützung der Anwendung,
- interaktive Authentifizierung.

---

**Fall E – Direkter Zugriff funktioniert, Proxyzugriff nicht**

Prüfen:

- Proxyfilter,
- Zielkategorie,
- Proxy-DNS,
- Authentifizierung,
- CONNECT-Erlaubnis,
- TLS-Inspection,
- Proxyprotokolle.

---

**Fall F – Reverse Proxy liefert 502**

Prüfen:

- Backenddienst,
- Backendname,
- Backendport,
- HTTP gegenüber HTTPS,
- Firewall,
- Container-Netzwerk,
- ungültige Backendantwort,
- TLS-Zertifikat des Backends.

---

**Fall G – Reverse Proxy liefert 504**

Prüfen:

- Backendantwortzeit,
- Datenbank oder externe API,
- Upstreamtimeout,
- Paketverlust,
- DNS-Verzögerung,

- ausgelastetes Backend.

---

**Fall H – Endlose Umleitung**

Prüfen:

- `X-Forwarded-Proto`,
- HTTPS-Erkennung der Anwendung,
- HTTP-zu-HTTPS-Regeln,
- doppelten Redirect,
- Basis-URL der Anwendung,
- Hostheader.

---

## 32. Keine vorschnellen Proxyänderungen durchführen

Nicht unkontrolliert:

- Systemproxy zurücksetzen,
- PAC-URL entfernen,
- WPAD-DNS-Einträge anlegen,
- Proxy-Authentifizierung deaktivieren,
- TLS-Inspection umgehen,
- Zertifikatsprüfung dauerhaft abschalten,
- Unternehmenszertifikate löschen,
- beliebige Ziele auf die Bypassliste setzen,
- Forwarded-Header ungeprüft vertrauen,
- Proxyprotokolle mit Zugangsdaten weitergeben,
- Reverse Proxy neu laden, bevor die Konfiguration geprüft wurde.

**Sicheres Vorgehen:**

1. Istkonfiguration dokumentieren.
2. Verwaltungsquelle bestimmen.
3. Betroffene Anwendung und Benutzerkontext feststellen.
4. Direkten und proxied Datenfluss getrennt prüfen.
5. Proxy- und Backendprotokolle zeitlich zuordnen.
6. Ursache nachweisen.
7. Änderung genehmigen lassen.
8. Kleinste erforderliche Änderung durchführen.
9. Funktion und Sicherheit prüfen.
10. Rückfallmöglichkeit dokumentieren.

---

## 33. Systematischer Diagnoseablauf

| Schritt | Prüfung | Leitfrage |
|---|---|---|
| 1 | Anwendung und Benutzerkontext bestimmen | Wer baut die Verbindung tatsächlich auf? |
| 2 | Ziel-URL dokumentieren | Welcher Host, Port und welches Protokoll werden verwendet? |
| 3 | Proxyart bestimmen | Forward Proxy, PAC, transparent oder Reverse Proxy? |
| 4 | Konfigurationsquelle ermitteln | System, Anwendung, Richtlinie, Variable oder MDM? |
| 5 | Proxyhostname auflösen | Wird die erwartete Proxyadresse verwendet? |
| 6 | Proxyport prüfen | Ist der Proxy transportseitig erreichbar? |
| 7 | PAC- beziehungsweise WPAD-Auswahl prüfen | Welcher Proxy wird für diese URL gewählt? |
| 8 | Umgehungsliste prüfen | Soll das Ziel direkt oder über Proxy erreicht werden? |
| 9 | Proxytest durchführen | Welche Antwort liefert der Proxy? |
| 10 | Authentifizierung prüfen | Wird 407 oder ein anderes Verfahren verwendet? |
| 11 | Direkten Vergleichstest durchführen | Liegt der Fehler nur im Proxyweg? |
| 12 | Zertifikatskette prüfen | Ist TLS-Inspection oder ein Vertrauensproblem beteiligt? |
| 13 | Reverse-Proxy-Frontend prüfen | Erreicht der Client den Reverse Proxy? |
| 14 | Backend vom Proxy aus prüfen | Kann der Proxy das Backend erreichen? |
| 15 | Header und Protokoll prüfen | Stimmen Host, Schema und Forwarded-Header? |
| 16 | Protokolle vergleichen | Welche Komponente erzeugt den Fehler? |
| 17 | Paketmitschnitt erstellen | An welcher Verbindung scheitert der Datenfluss? |
| 18 | Minimale Änderung planen | Welche konkrete Korrektur stellt den Sollzustand her? |
| 19 | Nachprüfung | Funktionieren Anwendung, Authentifizierung und TLS? |

| Schritt | Prüfung | Leitfrage |
|---|---|---|
| 20 | Dokumentation | Sind Ursache, Änderung und Rückfallplan festgehalten? |

## 34. Kompakte Befehlstabelle

| Aufgabe | Windows | Linux | macOS |
|---|---|---|---|
| Proxyvariablen | `[RO][SENS] Get-ChildItem Env: \| Where-Object Name -Match 'proxy'` | `[RO][SENS] env \| grep -i proxy` | `[RO][SENS] env \| grep -i proxy` |
| WinHTTP-Proxy | `[RO][SENS] netsh winhttp show proxy` | Nicht zutreffend | Nicht zutreffend |
| Erweiterter WinHTTP-Proxy | `[RO][SENS] netsh winhttp show advproxy` | Nicht zutreffend | Nicht zutreffend |
| Benutzerproxy | `[RO][SENS] Get-ItemProperty "HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings"` | Abhängig von Desktop und Anwendung | `[RO][SENS] scutil --proxy` |
| HTTP-Proxy | Abhängig von Anwendung | Über Umgebungs- oder Anwendungskonfiguration | `[RO][SENS] networksetup -getwebproxy "Wi-Fi"` |
| HTTPS-Proxy | Abhängig von Anwendung | Über Umgebungs- oder Anwendungskonfiguration | `[RO][SENS] networksetup -getsecurewebproxy "Wi-Fi"` |
| PAC-URL | Registry beziehungsweise WinHTTP prüfen | Abhängig von Desktop und Anwendung | `[RO][SENS] networksetup -getautoproxyurl "Wi-Fi"` |
| Proxyhostname auflösen | `[TEST] Resolve-DnsName PROXY` | `[TEST] getent ahosts PROXY` | `[TEST] dscacheutil -q host -a name PROXY` |
| Proxyport prüfen | `[TEST] Test-NetConnection PROXY -Port PORT` | `[TEST] nc -vz -w 5 PROXY PORT` | `[TEST] nc -vz -w 5 PROXY PORT` |
| Expliziter Proxytest | `[TEST][SENS] curl.exe -v --proxy http://PROXY:PORT URL` | `[TEST][SENS] curl -v --proxy http://PROXY:PORT URL` | `[TEST][SENS] curl -v --proxy http://PROXY:PORT URL` |
| Proxy umgehen | `[TEST][SENS] curl.exe -v --noproxy "*" URL` | `[TEST][SENS] curl -v --noproxy "*" URL` | `[TEST][SENS] curl -v --noproxy "*" URL` |
| Dienstumgebung | Windows-Dienstkontext getrennt prüfen | `[RO][PRIV][SENS] sudo systemctl show DIENST --property=Environment` | Abhängig von LaunchDaemon und Anwendung |
| Reverse-Proxy-Backend testen | `[TEST] Test-NetConnection BACKEND -Port PORT` | `[TEST] nc -vz -w 5 BACKEND PORT` | `[TEST] nc -vz -w 5 BACKEND PORT` |
| NGINX-Konfiguration | Falls installiert: `[RO] nginx -t` | `[RO][PRIV][SENS] sudo nginx -t` | Falls installiert: `[RO][PRIV][SENS] sudo nginx -t` |
| Containerprotokolle | `[RO][SENS] docker logs --tail 100 CONTAINER` | `[RO][SENS] docker logs --tail 100 CONTAINER` | `[RO][SENS] docker logs --tail 100 CONTAINER` |

`PROXY`, `PORT`, `URL`, `DIENST`, `BACKEND` und `CONTAINER` müssen durch die Werte des konkreten Störungsfalls ersetzt werden.

---

## 35. Dokumentationsvorlage für Proxyfehler

```
Störung:
Zeitpunkt:
Zeitzone:
Benutzer:
Client:
Client-IP:
Betriebssystem:
Betroffene Anwendung:
Anwendungsversion:

Ziel-URL:
Zielhostname:
Ziel-IP:
Zielport:
HTTP oder HTTPS:
Fehlermeldung:
HTTP-Statuscode:

Proxyart:
Proxyhostname:
Proxy-IP:
Proxyport:
Konfigurationsquelle:
PAC-URL:
WPAD verwendet:
Umgehungsliste:
Proxy-Authentifizierung:
TLS-Inspection:

WinHTTP-Konfiguration:
Benutzerkonfiguration:
Umgebungsvariablen:
```

Anwendungseigene Konfiguration:
Gruppenrichtlinie oder MDM:

Proxy-DNS erfolgreich:
Proxyport erreichbar:
Expliziter Proxytest:
Direkter Vergleichstest:
Zertifikatsaussteller:
Proxyprotokolleintrag:

Reverse Proxy beteiligt:
Frontend erreichbar:
Backendhostname:
Backend-IP:
Backendport:
Backendprotokoll:
Backend vom Proxy erreichbar:
Backenddienst aktiv:
Proxy-Fehlerprotokoll:
Backend-Fehlerprotokoll:
Forwarded-Header geprüft:

Paketmitschnitt vorhanden:
Festgestellte Ursache:
Genehmigte Änderung:
Rückfallplan:
Ergebnis der Nachprüfung:

---

## 36. Kontrollfragen nach der Diagnose

- Welche Anwendung baut die Verbindung auf?
- Unter welchem Benutzer- oder Dienstkonto läuft sie?
- Wird ein Forward Proxy, Reverse Proxy oder beides verwendet?
- Welche Proxykonfiguration verwendet genau diese Anwendung?
- Stimmen Proxyhostname und Proxyport?
- Kann der Proxyname aufgelöst werden?
- Ist der Proxyport erreichbar?
- Wird eine PAC-Datei oder WPAD verwendet?

- Welche Proxyentscheidung gilt für die konkrete URL?
- Soll das Ziel laut Umgehungsliste direkt erreicht werden?
- Existieren widersprüchliche Proxyvariablen?
- Wurde ein Test mit ausdrücklich gesetztem Proxy durchgeführt?
- Wurde ein genehmigter direkter Vergleichstest durchgeführt?
- Verlangt der Proxy eine Authentifizierung?
- Kann die Anwendung das Verfahren unterstützen?
- Wird TLS-Inspection eingesetzt?
- Verwendet die Anwendung den richtigen Zertifikatsspeicher?
- Erreicht der Client den Reverse Proxy?
- Kann der Reverse Proxy das Backend auflösen und erreichen?
- Stimmen Backendprotokoll und Backendport?
- Werden Host- und Forwarded-Header korrekt übermittelt?
- Wurde der HTTP-Statuscode anhand der Protokolle einer Komponente zugeordnet?
- Wurden Client-, Proxy- und Backendzeitstempel verglichen?
- Wurde die Konfiguration nicht vorschnell zurückgesetzt?
- Wurde die Funktion nach einer Änderung erneut vollständig geprüft?

---

## 37. Quellen und weiterführende Dokumentation

- Microsoft Learn – `netsh winhttp`:
  https://learn.microsoft.com/de-de/windows-server/administration/windows-commands/netsh-winhttp
- Microsoft Learn – WinHTTP AutoProxy Support:
  https://learn.microsoft.com/windows/win32/winhttp/winhttp-autoproxy-support
- Microsoft Learn – WPAD-Prozess:
  https://learn.microsoft.com/windows-server/networking/automatic-tunneling/disable-http-proxy-auth-features
- Apple – `networksetup`-Handbuch:
  Auf dem Mac lokal abrufbar mit `man networksetup`
- Apple – Netzwerk-Proxyeinstellungen auf dem Mac:
  https://support.apple.com/de-de/guide/mac-help/mchlp2591/mac
- curl – Proxy-Unterstützung:
  https://curl.se/docs/manpage.html#-x
- curl – Proxy-Umgebungsvariablen:
  https://curl.se/libcurl/c/libcurl-env.html
- curl – Alles rund um Proxys:
  https://everything.curl.dev/usingcurl/proxies/index.html
- MDN – Proxy-Autokonfigurationsdatei:
  https://developer.mozilla.org/docs/Web/HTTP/Guides/Proxy_servers_and_tunneling/Proxy_Auto-Configuration_PAC_file
- MDN – HTTP-Statuscode 407:
  https://developer.mozilla.org/docs/Web/HTTP/Reference/Status/407

- MDN – HTTP-Statuscode 502: https://developer.mozilla.org/docs/Web/HTTP/Reference/Status/502
- MDN – HTTP-Statuscode 504: https://developer.mozilla.org/docs/Web/HTTP/Reference/Status/504
- NGINX – Reverse Proxy: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/
- NGINX – Modul `ngx_http_proxy_module`: https://nginx.org/en/docs/http/ngx_http_proxy_module.html
- RFC 9110 – HTTP Semantics: https://www.rfc-editor.org/rfc/rfc9110.html
- RFC 7239 – Forwarded HTTP Extension: https://www.rfc-editor.org/rfc/rfc7239.html

---