3.10 Firewall- und Paketfilterfehler analysieren

Firewalls können Verbindungen erlauben, verwerfen, ausdrücklich ablehnen, protokollieren oder an andere Ziele weiterleiten. Eine Anwendung kann korrekt laufen und auf dem richtigen Port lauschen, während der Zugriff trotzdem durch einen lokalen oder zentralen Paketfilter verhindert wird.

Die zentrale Frage dieser Seite lautet:

> Erreicht der Datenverkehr das Zielsystem, und welche Firewallregel entscheidet tatsächlich über das Paket?

---

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 die Konfiguration
[DISRUPT]	Der Befehl kann Verbindungen oder Dienste beeinträchtigen

Firewallregeln enthalten häufig interne IP-Adressen, Servernamen, Anwendungen, Netzwerkbereiche und Sicherheitsrichtlinien. Entsprechende Ausgaben sind vertraulich zu behandeln.

---

2. Firewallfehler nicht vorschnell vermuten

Die Meldung „Das ist bestimmt die Firewall“ ist zunächst nur eine Hypothese. Vor einer Änderung müssen mindestens folgende Punkte geprüft werden:

1. Wurde der richtige Hostname verwendet?

2. Wurde die erwartete IP-Adresse aufgelöst?
3. Ist eine gültige Route zum Ziel vorhanden?
4. Läuft der benötigte Dienst?
5. Lauscht der Dienst auf dem erwarteten Port und Protokoll?
6. Lauscht der Dienst an der richtigen IP-Adresse?
7. Funktioniert der Zugriff lokal auf dem Server?
8. Funktioniert der Zugriff aus einem anderen Netzsegment?
9. Erreicht das Paket das Zielsystem?
10. Wird eine Blockierung protokolliert?

Wichtig:

> Timeout
>
> ≠
>
> sicherer Nachweis einer Firewallblockierung

Ein Timeout kann ebenfalls durch Paketverlust, falsches Routing, einen fehlenden Rückweg, eine falsche Zieladresse, NAT-Fehler oder einen ausgefallenen Host entstehen.

---

3. DROP, REJECT und ALLOW unterscheiden

Aktion	Verhalten	Typische Beobachtung
`ALLOW` beziehungsweise `ACCEPT`	Passender Verkehr wird zugelassen	Verbindung kann aufgebaut werden
`DROP`	Paket wird ohne Antwort verworfen	Client wartet häufig bis zum Timeout
`REJECT`	Paket wird abgelehnt und eine Fehlermeldung zurückgesendet	Client erhält häufig schnell einen Fehler
`LOG`	Paketinformationen werden protokolliert	Allein keine endgültige Erlaubnis oder Blockierung
`DNAT` beziehungsweise Portweiterleitung	Zieladresse oder Zielport wird verändert	Client erreicht möglicherweise ein internes Ziel
`SNAT` beziehungsweise Masquerading	Quelladresse wird verändert	Ziel erkennt die Adresse des NAT-Systems
`RETURN`	Verarbeitung kehrt zur aufrufenden Regelkette zurück	Wirkung hängt von weiteren Regeln ab

Bei TCP kann eine aktive Ablehnung beispielsweise durch ein TCP-RST sichtbar werden. Bei anderen Fällen kann eine ICMP- oder ICMPv6-Fehlermeldung zurückgegeben werden.

Das genaue Verhalten hängt vom Betriebssystem, Regelwerk und eingesetzten Sicherheitsprodukt ab.

---

4. Prüfen, welche Firewall tatsächlich verwendet wird

Auf einem System können mehrere Verwaltungsebenen vorhanden sein:

- native Betriebssystem-Firewall,
- `firewalld`,
- `nftables`,
- `iptables`,
- `ufw`,
- Sicherheitssoftware eines Drittanbieters,
- Endpoint-Detection-and-Response-System,
- Container-Firewallregeln,
- Hypervisor-Firewall,
- zentrale Netzwerk-Firewall,
- Router-ACL,
- Cloud-Sicherheitsgruppe,
- VPN- oder Zero-Trust-Client.

Eine Oberfläche wie `ufw` oder `firewalld` verwaltet im Hintergrund das eigentliche Paketfiltersystem. Regeln sollten deshalb nicht unkoordiniert über mehrere Werkzeuge verändert werden.

Betriebssystem	Typische lokale Filter
Windows	Windows Defender Firewall, IPsec-Regeln, Sicherheitssoftware
Linux	nftables, firewalld, ufw, iptables und Sicherheitssoftware
macOS	Anwendungsfirewall, Packet Filter `pf` und Sicherheitssoftware

Grundregel:

> Zuerst die aktive Verwaltungs- und Richtlinienebene identifizieren, danach Regeln bewerten.

---

5. Datenfluss und Prüfrichtung festlegen

Vor der Regelsuche muss der betroffene Datenfluss eindeutig beschrieben werden.

```
Client-IP:Quellport
        ↓
Transportprotokoll
        ↓
Ziel-IP:Zielport
        ↓
Serveranwendung
```

Zu dokumentieren sind:

Merkmal	Beispiel
Richtung	Client zum Server
Adressfamilie	IPv4
Transportprotokoll	TCP
Quelladresse	`192.0.2.25`
Quellport	dynamisch, beispielsweise `53144`
Zieladresse	`192.0.2.53`
Zielport	`443`
Schnittstelle	`Ethernet` beziehungsweise `eth0`
Netzwerkprofil oder Zone	`Domain` beziehungsweise `public`
Anwendung	Webserver
Zeitpunkt	Mit Zeitzone und Sekunden

Eingehend und ausgehend sind standortabhängig:

```
Client: ausgehende Verbindung
Server: eingehende Verbindung
```

Bei Routern, NAT-Systemen und Firewalls zwischen Netzen kann zusätzlich die Weiterleitungsrichtung betroffen sein.

6. Windows-Firewallstatus und aktive Profile prüfen

Windows verwendet die Profile:

- `Domain`
- `Private`
- `Public`

Je nach aktiver Netzwerkverbindung können unterschiedliche Profile und Regeln gelten.

Aktives Netzwerkprofil prüfen:

```
[RO] Get-NetConnectionProfile |
    Format-Table InterfaceAlias, Name, NetworkCategory, IPv4Connectivity, IPv6Connectivity
```

Status aller Firewallprofile:

```
[RO] Get-NetFirewallProfile |
    Format-Table Name, Enabled, DefaultInboundAction, DefaultOutboundAction
```

Ausführliche Profileinstellungen:

```
[RO] Get-NetFirewallProfile |
    Format-List Name, Enabled, DefaultInboundAction, DefaultOutboundAction,
                AllowInboundRules, AllowLocalFirewallRules,
                AllowLocalIPsecRules, NotifyOnListen,
                LogAllowed, LogBlocked, LogFileName, LogMaxSizeKilobytes
```

Klassische Statusanzeige:

```
[RO] netsh advfirewall show allprofiles
```

Zu kontrollieren sind:

- Welches Netzwerkprofil ist aktiv?
- Ist die Firewall für dieses Profil aktiviert?
- Was ist die Standardaktion für eingehenden Verkehr?
- Was ist die Standardaktion für ausgehenden Verkehr?
- Dürfen lokale Regeln angewendet werden?
- Werden die Einstellungen durch Gruppenrichtlinien verwaltet?

- Ist die Protokollierung zugelassener oder blockierter Pakete aktiviert?

Typischer Fehler:

```
Regel gilt nur für Profil Private
↓
Netzwerk wurde als Public erkannt
↓
Regel greift nicht
```

Das Netzwerkprofil darf nicht allein zur Fehlerbehebung verändert werden. Zuerst ist zu prüfen, warum Windows das Netzwerk entsprechend eingestuft hat und welche Unternehmensrichtlinie gelten soll.

---

7. Windows-Firewallregeln untersuchen

Alle aktivierten Regeln anzeigen:

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

Aktive Blockierungsregeln anzeigen:

```
[RO][SENS] Get-NetFirewallRule -Enabled True -Action Block |
    Format-Table DisplayName, Direction, Profile
```

Aktive eingehende Erlaubnisregeln anzeigen:

```
[RO][SENS] Get-NetFirewallRule -Enabled True -Direction Inbound -Action Allow |
    Format-Table DisplayName, Profile
```

Bestimmte Regel ausführlich anzeigen:

```
[RO][SENS] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Format-List *
```

Windows speichert Port-, Adress-, Programm-, Dienst- und Schnittstellenbedingungen in zugehörigen Filterobjekten. `Get-NetFirewallRule` allein zeigt deshalb nicht alle Bedingungen einer Regel.

Portbedingungen einer Regel:

```
[RO] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Get-NetFirewallPortFilter |
    Format-List *
```

Adressbedingungen einer Regel:

```
[RO][SENS] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Get-NetFirewallAddressFilter |
    Format-List *
```

Anwendungsbedingung einer Regel:

```
[RO][SENS] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Get-NetFirewallApplicationFilter |
    Format-List *
```

Dienstbedingung einer Regel:

```
[RO] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Get-NetFirewallServiceFilter |
    Format-List *
```

Schnittstellenbedingung einer Regel:

```
[RO] Get-NetFirewallRule -DisplayName "NAME DER REGEL" |
    Get-NetFirewallInterfaceFilter |
    Format-List *
```

Regeln für lokalen TCP-Port 443 suchen:

```
[RO][SENS] Get-NetFirewallPortFilter -Protocol TCP |
    Where-Object LocalPort -eq "443" |
    Get-NetFirewallRule |
    Format-Table DisplayName, Enabled, Direction, Action, Profile
```

Regeln für lokalen UDP-Port 53 suchen:

```
[RO][SENS] Get-NetFirewallPortFilter -Protocol UDP |
    Where-Object LocalPort -eq "53" |
    Get-NetFirewallRule |
    Format-Table DisplayName, Enabled, Direction, Action, Profile
```

Regeln für eine bestimmte Anwendung suchen:

```
[RO][SENS] Get-NetFirewallApplicationFilter |
    Where-Object Program -like "*\PROGRAMM.exe" |
    Get-NetFirewallRule |
    Format-Table DisplayName, Enabled, Direction, Action, Profile
```

Der Programmpfad muss durch den tatsächlichen Pfad beziehungsweise ein eindeutiges Suchmuster ersetzt werden.

---

8. Windows-Regeln vollständig beurteilen

Bei einer vermeintlich passenden Regel müssen alle Bedingungen kontrolliert werden:

Bedingung	Kontrollfrage
`Enabled`	Ist die Regel aktiviert?
`Direction`	Gilt sie eingehend oder ausgehend?
`Action`	Erlaubt oder blockiert sie?
`Profile`	Gilt sie für das aktive Profil?
`Protocol`	TCP, UDP, ICMPv4 oder ICMPv6?
`LocalPort`	Stimmt der lokale Port?
`RemotePort`	Ist ein bestimmter entfernter Port vorgegeben?

Bedingung	Kontrollfrage
`LocalAddress`	Gilt sie für die verwendete lokale Adresse?
`RemoteAddress`	Ist die Clientadresse oder ihr Netz erlaubt?
`Program`	Stimmt der ausführbare Programmpfad?
`Service`	Ist die Regel an einen bestimmten Dienst gebunden?
`InterfaceType`	Gilt sie für LAN, WLAN oder Remotezugriff?
`EdgeTraversalPolicy`	Ist spezieller eingehender Verkehr betroffen?
Richtlinienquelle	Stammt sie lokal, aus Gruppenrichtlinien oder MDM?

Richtlinienquelle anzeigen:

```
[RO][SENS] Get-NetFirewallRule -PolicyStore ActiveStore |
    Select-Object DisplayName, Enabled, Direction, Action, Profile,
                  PolicyStoreSourceType, PolicyStoreSource |
    Format-Table -AutoSize
```

`ActiveStore` repräsentiert die wirksame Zusammenführung der angewendeten Richtlinien. Eine lokal sichtbare Regel kann durch zentral verwaltete Vorgaben eingeschränkt sein.

Wichtig: Eine scheinbar passende Erlaubnisregel beweist nicht automatisch, dass keine andere wirksame Regel oder Sicherheitskomponente den Datenverkehr blockiert.

---

9. Windows-Firewallprotokollierung prüfen

Konfigurierte Protokolldateien und Einstellungen anzeigen:

```
[RO] Get-NetFirewallProfile |
    Select-Object Name, LogAllowed, LogBlocked, LogFileName, LogMaxSizeKilobytes |
    Format-List
```

Der konkrete Speicherort muss aus `LogFileName` übernommen werden. Er sollte nicht angenommen oder fest im Diagnoseablauf vorgegeben werden.

Konfigurierte Protokolldatei lesen:

```
[RO][PRIV][SENS] Get-Content "PFAD AUS LogFileName" -Tail 100
```

Datei fortlaufend beobachten:

```
[RO][PRIV][SENS] Get-Content "PFAD AUS LogFileName" -Tail 20 -Wait
```

Während der Beobachtung wird der fehlgeschlagene Verbindungstest erneut ausgeführt. Dabei müssen Zeitpunkt, Quelladresse, Zieladresse, Protokoll und Port abgeglichen werden.

Ereignisprotokolle zur Firewall suchen:

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

Anschließend kann ein tatsächlich vorhandenes Protokoll gezielt abgefragt werden:

```
[RO][PRIV][SENS] Get-WinEvent -LogName "ERMITTELTER PROTOKOLLNAME" -MaxEvents 50
```

Nicht jede Paketblockierung erscheint automatisch im Windows-Ereignisprotokoll. Die erforderliche Überwachung beziehungsweise Protokollierung muss zuvor aktiviert und durch die Unternehmensrichtlinie erlaubt sein.

---

10. Linux-Firewallsystem identifizieren

Vorhandene Werkzeuge ermitteln:

```
[RO] command -v firewall-cmd
[RO] command -v nft
[RO] command -v ufw
[RO] command -v iptables
```

Aktive Dienste prüfen:

```
[RO] systemctl is-active firewalld
[RO] systemctl is-active nftables
[RO] systemctl is-active ufw
```

Eine Ausgabe wie `inactive`, `failed` oder `unknown` muss im Kontext der jeweiligen Distribution bewertet werden. Beispielsweise kann ein nftables-Regelwerk aktiv sein, obwohl kein dauerhaft laufender `nftables`-Dienst angezeigt wird.

Geladene nftables-Regeln prüfen:

```
[RO][PRIV][SENS] sudo nft list ruleset
```

iptables-Regeln prüfen:

```
[RO][PRIV][SENS] sudo iptables -S
[RO][PRIV][SENS] sudo ip6tables -S
```

iptables-Regeln mit Zählern:

```
[RO][PRIV][SENS] sudo iptables -L -n -v --line-numbers
[RO][PRIV][SENS] sudo ip6tables -L -n -v --line-numbers
```

Auf modernen Linux-Systemen kann der Befehl `iptables` ein Kompatibilitätsfrontend für nftables sein. Deshalb muss geprüft werden, welche Verwaltungsebene die Distribution tatsächlich verwendet.

---

11. firewalld analysieren

`firewalld` arbeitet mit Zonen. Schnittstellen oder Quellnetze werden einer Zone zugeordnet, für die Dienste, Ports und weitere Regeln gelten.

Status prüfen:

```
[RO] firewall-cmd --state
```

Standardzone anzeigen:

```
[RO] firewall-cmd --get-default-zone
```

Aktive Zonen und zugehörige Schnittstellen anzeigen:

```
[RO] firewall-cmd --get-active-zones
```

Zone einer bestimmten Schnittstelle ermitteln:

```
[RO] firewall-cmd --get-zone-of-interface=eth0
```

`eth0` muss durch die tatsächliche Schnittstelle ersetzt werden.

Komplette Laufzeitkonfiguration einer Zone anzeigen:

```
[RO][SENS] firewall-cmd --zone=public --list-all
```

Alle Zonen anzeigen:

```
[RO][SENS] firewall-cmd --list-all-zones
```

Freigegebene Dienste:

```
[RO] firewall-cmd --zone=public --list-services
```

Freigegebene Ports:

```
[RO] firewall-cmd --zone=public --list-ports
```

Prüfen, ob TCP-Port 443 in der Laufzeitkonfiguration freigegeben ist:

```
[RO] firewall-cmd --zone=public --query-port=443/tcp
```

Prüfen, ob der HTTPS-Dienst freigegeben ist:

```
[RO] firewall-cmd --zone=public --query-service=https
```

Rich Rules anzeigen:

```
[RO][SENS] firewall-cmd --zone=public --list-rich-rules
```

Policies anzeigen:

```
[RO][SENS] firewall-cmd --get-policies
```

Bei jeder Prüfung muss die tatsächlich aktive Zone verwendet werden. Die Zone `public` ist nur ein Beispiel.

---

12. firewalld: Laufzeit- und dauerhafte Konfiguration vergleichen

`firewalld` trennt zwischen:

- **Runtime-Konfiguration:** aktuell im laufenden System wirksam
- **Permanent-Konfiguration:** für erneutes Laden beziehungsweise Neustart gespeichert

Aktuelle Laufzeitkonfiguration:

```
[RO][SENS] firewall-cmd --zone=public --list-all
```

Dauerhafte Konfiguration:

```
[RO][SENS] firewall-cmd --permanent --zone=public --list-all
```

Typische Abweichungen:

Situation	Folge
Port nur zur Laufzeit freigegeben	Freigabe verschwindet nach Reload oder Neustart
Port nur dauerhaft eingetragen	Freigabe ist noch nicht zwingend zur Laufzeit aktiv
Falsche Zone bearbeitet	Regel existiert, gilt aber nicht für die betroffene Schnittstelle
Dienstname freigegeben, aber andere Portdefinition erwartet	Tatsächlicher Dienstport kann abweichen
Schnittstellenzuordnung geändert	Anderes Zonenregelwerk wird wirksam

Eine Abweichung ist nicht automatisch ein Fehler. Sie muss mit dem vorgesehenen Sollzustand verglichen werden.

---

13. ufw analysieren

Status und Standardrichtlinien anzeigen:

```
[RO][PRIV][SENS] sudo ufw status verbose
```

Regeln nummeriert anzeigen:

```
[RO][PRIV][SENS] sudo ufw status numbered
```

Erweiterte Rohdarstellung anzeigen:

```
[RO][PRIV][SENS] sudo ufw show raw
```

Zu kontrollieren sind:

- Ist `ufw` aktiv?
- Welche Standardaktion gilt eingehend?
- Welche Standardaktion gilt ausgehend?
- Ist der richtige Port freigegeben?
- Gilt die Regel für TCP oder UDP?
- Ist die Regel auf eine Quelladresse beschränkt?
- Existiert eine separate IPv6-Regel?
- Gilt die Regel für die richtige Schnittstelle?
- Gibt es vorhergehende Ablehnungsregeln?

Beispiel einer begrenzten Regelbeschreibung:

```
TCP-Port 443
nur aus 192.0.2.0/24
eingehend
über die vorgesehene Schnittstelle
```

Eine Ausgabe wie `443 ALLOW Anywhere` muss zusätzlich hinsichtlich IPv4, IPv6 und der tatsächlich aktiven Backend-Regeln bewertet werden.

---

14. nftables-Regelwerk untersuchen

Gesamtes geladenes Regelwerk anzeigen:

```
[RO][PRIV][SENS] sudo nft list ruleset
```

Tabellen anzeigen:

```
[RO][PRIV] sudo nft list tables
```

Regelwerk mit Handles anzeigen:

```
[RO][PRIV][SENS] sudo nft -a list ruleset
```

Regelwerk mit Zählern und Handles beobachten:

```
[RO][PRIV][SENS] sudo nft -a list ruleset
```

Bei nftables sind insbesondere zu prüfen:

Bestandteil	Bedeutung
Familie	`ip`, `ip6`, `inet`, `arp`, `bridge` oder `netdev`
Tabelle	Logische Sammlung von Chains und Regeln
Base Chain	An einen Netfilter-Hook gebundene Kette
Hook	Beispielsweise `input`, `output` oder `forward`
Priority	Reihenfolge gegenüber anderen Base Chains
Policy	Standardaktion der Base Chain
Match	Bedingung wie Adresse, Port, Protokoll oder Schnittstelle
Verdict	Beispielsweise `accept`, `drop`, `reject`, `jump` oder `return`
Counter	Anzahl passender Pakete und Bytes
Handle	Interne Kennung einer Regel

Wichtige Diagnosefrage:

> Steigt der Zähler der vermuteten Regel, während der Verbindungstest ausgeführt wird?

Wenn der Zähler nicht steigt, kann dies bedeuten:

- Das Paket erreicht das System nicht.
- Das Paket durchläuft eine andere Chain.
- Eine vorherige Regel entscheidet bereits.
- Protokoll, Port oder Adresse stimmen nicht.
- IPv4 und IPv6 wurden verwechselt.
- Eine andere Firewallkomponente greift ein.

---

15. iptables-Regeln untersuchen

Filterregeln mit Zählern und Zeilennummern:

```
[RO][PRIV][SENS] sudo iptables -L -n -v --line-numbers
```

IPv6-Regeln:

```
[RO][PRIV][SENS] sudo ip6tables -L -n -v --line-numbers
```

Regeln in Befehlsschreibweise:

```
[RO][PRIV][SENS] sudo iptables -S
```

NAT-Tabelle:

```
[RO][PRIV][SENS] sudo iptables -t nat -L -n -v --line-numbers
```

Weitere häufig relevante Tabellen:

```
[RO][PRIV][SENS] sudo iptables -t mangle -L -n -v --line-numbers
[RO][PRIV][SENS] sudo iptables -t raw -L -n -v --line-numbers
```

Zu prüfen sind:

- Standardrichtlinie der Chain,
- Reihenfolge der Regeln,
- Quell- und Zieladressen,

- Eingangs- und Ausgangsschnittstelle,
- Transportprotokoll,
- Quell- und Zielport,
- Verbindungszustand,
- Sprung in benutzerdefinierte Chains,
- Paket- und Bytezähler,
- IPv4- und IPv6-Regeln,
- NAT-Regeln.

Bei iptables entscheidet häufig die erste zutreffende Regel, die ein endgültiges Ziel wie `ACCEPT`, `DROP` oder `REJECT` ausführt. Eine Erlaubnisregel unterhalb einer bereits passenden Blockierungsregel wird dann nicht erreicht.

---

16. macOS-Anwendungsfirewall prüfen

Die macOS-Anwendungsfirewall kontrolliert hauptsächlich eingehende Verbindungen anhand von Anwendungen und Diensten. Sie ist nicht mit dem Packet Filter `pf` gleichzusetzen.

Grafische Prüfung:

```
Systemeinstellungen
→ Netzwerk
→ Firewall
```

Je nach macOS-Version können die genaue Position und Bezeichnungen abweichen.

Status der Anwendungsfirewall:

```
[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getglobalstate
```

Prüfen, ob alle eingehenden Verbindungen blockiert werden:

```
[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getblockall
```

Stealth-Modus prüfen:

```
[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getstealthmode
```

Automatische Freigabe signierter Software prüfen:

```
[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getallowsigned
```

Konfigurierte Anwendungen anzeigen:

```
[RO][PRIV][SENS] sudo /usr/libexec/ApplicationFirewall/socketfilterfw --listapps
```

Zu kontrollieren sind:

- Ist die Anwendungsfirewall eingeschaltet?
- Werden alle eingehenden Verbindungen blockiert?
- Ist die betroffene Anwendung aufgeführt?
- Ist die Anwendung zugelassen oder blockiert?
- Hat sich der Programmpfad nach einem Update geändert?
- Wird der Dienst durch eine andere ausführbare Datei gestartet?
- Wird die Einstellung durch ein Konfigurationsprofil oder MDM verwaltet?

Die genaue Ausgabe von `socketfilterfw` kann zwischen macOS-Versionen variieren.

---

17. macOS Packet Filter pf prüfen

`pf` ist ein paketbasierter Filter und getrennt von der macOS-Anwendungsfirewall zu betrachten.

Status und grundlegende Informationen:

```
[RO][PRIV] sudo pfctl -s info
```

Geladene Filterregeln anzeigen:

```
[RO][PRIV][SENS] sudo pfctl -s rules
```

NAT-Regeln anzeigen:

```
[RO][PRIV][SENS] sudo pfctl -s nat
```

Alle verfügbaren pf-Informationen anzeigen:

```
[RO][PRIV][SENS] sudo pfctl -s all
```

Zustandstabelle anzeigen:

```
[RO][PRIV][SENS] sudo pfctl -s states
```

Regeln mit zusätzlichen Informationen anzeigen:

```
[RO][PRIV][SENS] sudo pfctl -vvs rules
```

Zu kontrollieren sind:

- Ist `pf` aktiviert?
- Welche Regeln und Anchors wurden geladen?
- Gibt es NAT- oder Weiterleitungsregeln?
- Stimmen Schnittstelle, Adressfamilie, Adresse, Protokoll und Port?
- Erhöhen sich Regelzähler während des Tests?
- Wird das Regelwerk durch VPN-, Virtualisierungs- oder Sicherheitssoftware ergänzt?

Nicht ausführen, solange keine ausdrücklich genehmigte Änderung vorgesehen ist:

```
pf aktivieren oder deaktivieren
Regelwerk neu laden
Zustandstabellen leeren
Anchors verändern
```

Solche Aktionen können bestehende Netzwerkverbindungen und Sicherheitsrichtlinien beeinflussen.

---

18. Host-Firewall oder Netzwerk-Firewall unterscheiden

Ein Vergleich mehrerer Messpunkte hilft, die blockierende Stelle einzugrenzen.

```
Client
  ↓
Client-Firewall
```

```
  ↓
Switch/VLAN
  ↓
Router oder zentrale Firewall
  ↓
Server-Firewall
  ↓
Serveranwendung
```

Beobachtung	Mögliche Schlussfolgerung
Paket verlässt den Client nicht	Client-Firewall, Anwendung oder lokales Routing prüfen
Paket verlässt Client, erreicht Server aber nicht	Netzwerkpfad, ACL, VPN, NAT oder zentrale Firewall prüfen
Paket erreicht Server, aber Anwendung erhält es nicht	Server-Firewall oder falsche Bindung prüfen
SYN erreicht Server, keine Antwort verlässt ihn	Listener, Server-Firewall oder lokales Routing prüfen
SYN-ACK verlässt Server, erreicht Client nicht	Rückweg oder Filter zwischen Server und Client prüfen
Verbindung funktioniert lokal, aber nicht entfernt	Bind-Adresse, Server-Firewall oder Netzwerkfilter prüfen
Nur ein Quellnetz ist betroffen	Quellnetz-, VLAN-, Zonen- oder Adressregel prüfen
Nur IPv6 ist betroffen	IPv6-Regeln und ICMPv6 gesondert prüfen

Ein gleichzeitiger Paketmitschnitt auf Client und Server ist wesentlich aussagekräftiger als ein einseitiger Porttest.

---

19. Paketmitschnitt zur Firewallanalyse verwenden

Linux – TCP-Port 443 auf allen Schnittstellen:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i any -nn \
    'host 192.0.2.25 and tcp port 443' \
    -w firewall-test.pcap
```

macOS – TCP-Port 443 auf einer konkreten Schnittstelle:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i en0 -nn \
    'host 192.0.2.25 and tcp port 443' \
```

```
    -w firewall-test.pcap
```

Windows mit Dumpcap – Schnittstellen ermitteln:

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

Windows – Mitschnitt auf der zuvor bestimmten Schnittstelle:

```
[TEST][PRIV][FILE][SENS] dumpcap -i 1 `
    -f "host 192.0.2.25 and tcp port 443" `
    -w firewall-test.pcapng
```

Die Schnittstellennamen und -nummern sind Beispiele und müssen vorher ermittelt werden.

Wireshark-Anzeigefilter:

Aufgabe	Filter
Gesamter Verkehr zwischen zwei IPv4-Systemen	`ip.addr == 192.0.2.25 && ip.addr == 192.0.2.53`
TCP-Port 443	`tcp.port == 443`
Erste TCP-SYN-Pakete	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
TCP-Reset	`tcp.flags.reset == 1`
ICMP-Fehler	`icmp`
ICMPv6-Fehler	`icmpv6`
Vermutete Wiederholungen	`tcp.analysis.retransmission`

Interpretation:

Paketmuster	Mögliche Bedeutung
Wiederholtes SYN ohne Antwort	DROP, Paketverlust, falscher Rückweg oder ausgefallenes Ziel
SYN gefolgt von RST/ACK	Port geschlossen oder aktive Ablehnung
ICMP administratively prohibited	Paketfilter oder Richtlinie lehnt Verkehr ausdrücklich ab
SYN und SYN-ACK sichtbar, ACK fehlt	Rückweg oder Clientseite prüfen
Vollständiger Handshake sichtbar	Firewall lässt den TCP-Aufbau grundsätzlich zu
Handshake erfolgreich, danach Abbruch	Anwendungs-, TLS- oder Sitzungsproblem möglich

Der Mitschnitt beweist nur, was am jeweiligen Messpunkt sichtbar war.

20. Firewallprotokolle zeitlich korrekt auswerten

Für eine zuverlässige Zuordnung werden folgende Werte benötigt:

- Datum und Uhrzeit
- Zeitzone
- Quelladresse
- Zieladresse
- Transportprotokoll
- Quellport
- Zielport
- Richtung
- Aktion
- Schnittstelle oder Zone
- Regelname beziehungsweise Regel-ID

Systemzeit prüfen:

Betriebssystem	Befehl
Windows	[RO] Get-Date -Format o
Linux	[RO] date --iso-8601=seconds
macOS	[RO] date "+%Y-%m-%dT%H:%M:%S%z"

Bei zentralen Firewalls können Zeitstempel in UTC gespeichert werden. Client-, Server- und Firewallzeit müssen deshalb einschließlich Zeitzone verglichen werden.

Empfohlener Testablauf:

1. Exakten Startzeitpunkt notieren.
2. Einen einzelnen reproduzierbaren Verbindungstest durchführen.
3. Exakten Endzeitpunkt notieren.
4. Protokolle auf dieses Zeitfenster beschränken.
5. Quell- und Zieladresse sowie Port abgleichen.
6. Regelname und Aktion dokumentieren.
7. Bei NAT sowohl ursprüngliche als auch übersetzte Adressen berücksichtigen.

21. Stateful Inspection und Verbindungszustände berücksichtigen

Moderne Firewalls arbeiten häufig zustandsorientiert. Sie verfolgen bestehende Verbindungen und unterscheiden beispielsweise:

- neue Verbindung,
- bestehende Verbindung,
- zugehörige Verbindung,
- ungültiger Zustand.

Dadurch kann Rückverkehr erlaubt sein, obwohl keine allgemeine eingehende Freigabe für den dynamischen Clientport existiert.

Linux – Conntrack-Werkzeug vorhanden?

```
[RO] command -v conntrack
```

Verbindungseinträge anzeigen, sofern das Werkzeug installiert ist:

```
[RO][PRIV][SENS] sudo conntrack -L
```

Nur TCP-Verbindungen zu Port 443 filtern:

```
[RO][PRIV][SENS] sudo conntrack -L -p tcp --dport 443
```

`conntrack` ist nicht auf jedem Linux-System standardmäßig installiert.

Mögliche Fehlerbilder:

- asymmetrisches Routing führt nur eine Richtung durch die zustandsorientierte Firewall,
- veraltete Zustandseinträge beeinflussen neue Verbindungen,
- NAT-Zustände stimmen nicht mit dem aktuellen Pfad überein,
- Pakete werden als ungültig bewertet,
- mehrere Firewalls sehen unterschiedliche Teile derselben Verbindung.

Zustandstabellen dürfen nicht ohne Freigabe geleert werden. Das kann zahlreiche bestehende Verbindungen unterbrechen.

---

22. ICMP und ICMPv6 nicht pauschal blockieren

ICMP beziehungsweise ICMPv6 wird nicht nur für `ping` verwendet. Es übermittelt wichtige Netzwerk- und Fehlermeldungen.

Dazu gehören unter anderem:

- Ziel nicht erreichbar,
- Fragmentierung erforderlich,
- Packet Too Big,
- Time Exceeded,
- Echo Request und Echo Reply,
- Router- und Neighbor-Discovery-Funktionen bei IPv6.

Eine zu weitgehende Blockierung kann zu schwer erkennbaren Fehlern führen:

```
Kleine Pakete funktionieren
↓
Größere Übertragungen bleiben hängen
↓
Erforderliche MTU-Fehlermeldung wird blockiert
↓
Path-MTU-Ermittlung funktioniert nicht richtig
```

Bei IPv6 sind bestimmte ICMPv6-Funktionen für den regulären Betrieb notwendig. „ICMP vollständig blockieren“ ist daher keine geeignete allgemeine Sicherheitsregel.

---

23. Typische Firewallfehler

Fehler	Wirkung
Falsches Windows-Profil	Regel ist vorhanden, aber nicht aktiv
Falsche firewalld-Zone	Regel gilt nicht für die betroffene Schnittstelle
TCP statt UDP freigegeben	Anwendung bleibt nicht erreichbar
UDP statt TCP freigegeben	TCP-Verbindung schlägt weiterhin fehl
Falscher lokaler oder entfernter Port	Regel passt nicht zum Datenfluss
Zu enges Quellnetz	Bestimmte Clients funktionieren, andere nicht
IPv4-Regel ohne IPv6-Entsprechung	Zugriff funktioniert nur über IPv4
Falscher Programmpfad	Anwendungsregel greift nach Update nicht mehr
Erlaubnisregel unterhalb einer Blockierungsregel	Vorherige Regel entscheidet bereits

Fehler	Wirkung
Runtime- und permanente Regeln weichen ab	Fehler erscheint nach Reload oder Neustart
Eingehend und ausgehend verwechselt	Falsche Richtung wurde freigegeben
INPUT und FORWARD verwechselt	Router- oder Containerverkehr wird falsch bewertet
Rückweg läuft über andere Firewall	Stateful Inspection kann Verbindung verwerfen
Lokale und zentrale Richtlinie widersprechen sich	Erwartete lokale Regel ist nicht wirksam
Containerport nicht veröffentlicht	Host-Firewallregel allein löst das Problem nicht
Sicherheitssoftware zusätzlich aktiv	Native Firewallregeln erklären das Verhalten nicht vollständig

---

24. Containerverkehr und Firewalls berücksichtigen

Containerverkehr kann zusätzliche Regelketten, Bridges, NAT-Regeln und Weiterleitungen verwenden.

```
Client
  ↓
Host-Firewall
  ↓
veröffentlichter Hostport
  ↓
NAT oder Container-Proxy
  ↓
Container-Netzwerk
  ↓
Containerport
```

Docker-Portzuordnungen prüfen:

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

Portzuordnung eines Containers:

```
[RO] docker port CONTAINERNAME
```

Docker-Netzwerke anzeigen:

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

Konkretes Netzwerk untersuchen:

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

Zu prüfen sind:

- Ist der Port überhaupt veröffentlicht?
- Ist er nur an `127.0.0.1` gebunden?
- Stimmen Hostport und Containerport?
- Lauscht die Anwendung im Container?
- Ist IP-Forwarding erforderlich und aktiv?
- Wird der Verkehr durch die `FORWARD`-Kette verarbeitet?
- Verwalten Docker und das Host-Firewallsystem gemeinsame Regeln?
- Existiert ein Reverse Proxy vor dem Container?

Firewallregeln, die von einer Containerplattform erzeugt wurden, dürfen nicht manuell verändert werden, ohne deren Verwaltungslogik und Auswirkungen zu berücksichtigen.

---

25. Sichere Gegenprobe ohne vollständiges Abschalten

Eine Firewall vollständig zu deaktivieren ist keine bevorzugte Diagnosemethode. Bessere Gegenproben sind:

1. Regelzähler vor und nach einem einzelnen Test vergleichen.
2. Blockierungsprotokoll aktivieren, sofern genehmigt.
3. Gleichzeitig auf Client und Server mitschneiden.
4. Eine eng begrenzte temporäre Testregel beantragen.
5. Nur eine definierte Quelladresse erlauben.
6. Nur den benötigten Zielport und das richtige Protokoll erlauben.
7. Testzeitraum begrenzen.
8. Änderung und Rückbau dokumentieren.
9. Regel nach dem Test entfernen oder deaktivieren.
10. Anschließend den ursprünglichen Zustand bestätigen.

Beispiel für den Umfang einer genehmigten Testregel:

```
Quelle:       192.0.2.25
Ziel:         192.0.2.53
Protokoll:    TCP
Zielport:     443
Richtung:     eingehend
Gültigkeit:   nur während des Wartungsfensters
Protokoll:    aktiv
```

Die konkreten Änderungsbefehle werden erst nach Prüfung des vorhandenen Regelwerks, der Verwaltungsplattform und des genehmigten Sollzustands erstellt.

---

26. Warum die Firewall nicht einfach deaktiviert werden sollte

Ein vollständiges Abschalten kann:

- das System ungeschützt erreichbar machen,
- gegen Unternehmensrichtlinien verstoßen,
- andere Sicherheitsfunktionen beeinträchtigen,
- den ursprünglichen Fehlerzustand verändern,
- zentrale Richtlinien umgehen oder Konflikte erzeugen,
- nicht beweisen, welche konkrete Regel verantwortlich war,
- zusätzliche Verbindungen zulassen, die den Test verfälschen,
- bei Remoteadministration zum Verbindungsabbruch führen.

Microsoft empfiehlt ausdrücklich, die Windows-Firewall nicht einfach zu deaktivieren. Stattdessen sollen gezielte Regeln und Diagnosefunktionen verwendet werden.

Nicht als allgemeiner Diagnoseschritt verwenden:

```
Windows-Firewall vollständig ausschalten
firewalld stoppen
ufw deaktivieren
nftables-Regelwerk leeren
iptables-Regeln leeren
pf deaktivieren
Sicherheitssoftware beenden
```

Diese Aktionen sind Änderungen mit potenziell weitreichenden Sicherheits- und Betriebsfolgen.

---

27. Systematischer Diagnoseablauf

Schritt	Prüfung	Leitfrage
1	Datenfluss definieren	Welche Quelle muss welches Ziel über welches Protokoll und welchen Port erreichen?
2	Fehler reproduzieren	Ist das Verhalten reproduzierbar und zeitlich dokumentiert?
3	DNS und Zieladresse prüfen	Wird wirklich das erwartete System angesprochen?
4	Routing prüfen	Verwendet das Paket den erwarteten Weg?
5	Serverdienst prüfen	Läuft der Dienst und lauscht er richtig?
6	Lokalen Test durchführen	Funktioniert der Dienst direkt auf dem Server?
7	Firewallsystem identifizieren	Welches Werkzeug und welche Richtlinie sind tatsächlich aktiv?
8	Profil, Zone oder Schnittstelle prüfen	Welcher Regelbereich gilt für den Verkehr?
9	Passende Regeln suchen	Stimmen Richtung, Protokoll, Port, Adressen und Anwendung?
10	Reihenfolge und Priorität prüfen	Entscheidet vorher bereits eine andere Regel?
11	Zähler und Protokolle beobachten	Trifft der Testverkehr auf die vermutete Regel?
12	Client- und Servermitschnitt vergleichen	An welcher Stelle verschwindet das Paket?
13	Zentrale Filter prüfen	Gibt es ACL, VPN-, Cloud- oder Netzwerk-Firewalls?
14	Eng begrenzte Änderung planen	Welche minimale Änderung stellt den Sollzustand her?
15	Änderung genehmigen und dokumentieren	Wer hat Umfang und Zeitraum freigegeben?
16	Nachprüfung durchführen	Funktioniert der Dienst und bleibt die Sicherheit erhalten?

---

28. Kompakte Befehlstabelle

Aufgabe	Windows	Linux	macOS
Aktives Netzwerkprofil oder Zone	`[RO] Get-NetConnectionProfile`	firewalld: `[RO] firewall-cmd --get-active-zones`	Anwendungsfirewall besitzt keine direkt vergleichbaren Netzwerkprofile
Firewallstatus	`[RO] Get-NetFirewallProfile`	firewalld: `[RO] firewall-cmd --state`	`[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getglobalstate`
Aktive Regeln	`[RO][SENS] Get-NetFirewallRule -Enabled True`	nftables: `[RO][PRIV][SENS] sudo nft list ruleset`	pf: `[RO][PRIV][SENS] sudo pfctl -s rules`
Blockierungsregeln	`[RO][SENS] Get-NetFirewallRule -Enabled True -Action Block`	Regelwerk nach `drop` beziehungsweise `reject` untersuchen	Anwendungsregeln: `[RO][PRIV][SENS] sudo /usr/libexec/ApplicationFirewall/socketfilterfw --listapps`
TCP-Portregel suchen	`[RO] Get-NetFirewallPortFilter -Protocol TCP`	firewalld: `[RO] firewall-cmd --zone=ZONE --query-port=443/tcp`	pf-Regeln: `[RO][PRIV][SENS] sudo pfctl -vvs rules`
UDP-Portregel suchen	`[RO] Get-NetFirewallPortFilter -Protocol UDP`	firewalld: `[RO] firewall-cmd --zone=ZONE --query-port=53/udp`	pf-Regeln: `[RO][PRIV][SENS] sudo pfctl -vvs rules`
Firewallprofile ausführlich	`[RO] Get-NetFirewallProfile | Format-List *`	Nicht direkt vergleichbar	`[RO] /usr/libexec/ApplicationFirewall/socketfilterfw --getblockall`
iptables-Regeln	Nicht zutreffend	`[RO][PRIV][SENS] sudo iptables -L -n -v --line-numbers`	Nicht standardmäßig verwendet
nftables-Regeln	Nicht zutreffend	`[RO][PRIV][SENS] sudo nft -a list ruleset`	Nicht zutreffend
ufw-Status	Nicht zutreffend	`[RO][PRIV][SENS] sudo ufw status verbose`	Nicht zutreffend
pf-Status	Nicht zutreffend	Nicht standardmäßig für alle Distributionen	`[RO][PRIV] sudo pfctl -s info`
TCP-Porttest	`[TEST] Test-NetConnection HOST -Port 443`	`[TEST] nc -vz -w 5 HOST 443`	`[TEST] nc -vz -w 5 HOST 443`
Paketmitschnitt	Dumpcap oder pktmon	`[TEST][PRIV][FILE][SENS] sudo tcpdump ...`	`[TEST][PRIV][FILE][SENS] sudo tcpdump ...`

`HOST`, `ZONE`, Schnittstellen, Adressen und Ports müssen durch die Werte des konkreten Störungsfalls ersetzt werden.

---

29. Dokumentationsvorlage für Firewallfehler

Störung:

Zeitpunkt:

Zeitzone:

Betroffener Standort:

Betroffener Benutzer oder Dienst:

Clienthostname:

Client-IP-Adresse:

Client-VLAN:

Client-Schnittstelle:

Zielhostname:

Ziel-IP-Adresse:

Zielport:

Transportprotokoll:

Adressfamilie: IPv4 / IPv6

Erwartete Anwendung:

Dienststatus:

Lokaler Listener:

Bind-Adresse:

Lokaler Funktionstest:

Entfernter Verbindungstest:

Fehlermeldung:

Aktive lokale Firewall:

Aktives Profil beziehungsweise aktive Zone:

Standardaktion eingehend:

Standardaktion ausgehend:

Vermutete Regel:

Regelrichtung:

Regelaktion:

Quellbeschränkung:

Zielbeschränkung:

Portbedingung:

Programmbedingung:

Richtlinienquelle:

Regelzähler vor dem Test:

Regelzähler nach dem Test:

Firewallprotokolleintrag:

Paket am Client sichtbar:

Paket am Server sichtbar:

Antwort am Server sichtbar:

Antwort am Client sichtbar:

Zentrale Firewall oder ACL:

NAT beteiligt:

Container oder Reverse Proxy beteiligt:

Festgestellte Ursache:

Genehmigte Änderung:

Rückfallplan:

Ergebnis der Nachprüfung:

30. Kontrollfragen nach der Diagnose

- Wurde der Datenfluss vollständig beschrieben?
- Wurde zwischen eingehendem, ausgehendem und weitergeleitetem Verkehr unterschieden?
- Wurde TCP, UDP, ICMP oder ICMPv6 korrekt bestimmt?
- Wurde die tatsächlich verwendete Zieladresse dokumentiert?
- Wurde geprüft, ob der Dienst lokal lauscht?
- Wurde das aktive Windows-Profil beziehungsweise die aktive Linux-Zone ermittelt?
- Wurde zwischen IPv4- und IPv6-Regeln unterschieden?
- Wurden Adress-, Port-, Anwendungs- und Dienstfilter geprüft?
- Wurde die Reihenfolge beziehungsweise Priorität der Regeln berücksichtigt?
- Wurden zentrale Richtlinien und Drittanbieterprodukte berücksichtigt?
- Wurde zwischen Runtime- und permanenter Konfiguration unterschieden?
- Wurden Regelzähler während eines reproduzierbaren Tests beobachtet?
- Wurden Client- und Servermitschnitt verglichen?
- Wurde ein Timeout nicht automatisch als Firewallbeweis gewertet?
- Wurde die Firewall nicht unnötig vollständig deaktiviert?
- Wurde eine Änderung auf das erforderliche Minimum begrenzt?
- Wurde der Sollzustand nach der Änderung erneut geprüft?
- Wurde die Änderung einschließlich Rückfallplan dokumentiert?

31. Quellen und weiterführende Dokumentation

- Microsoft Learn – Windows Firewall overview:
  https://learn.microsoft.com/windows/security/operating-system-security/network-security/windows-firewall/
- Microsoft Learn – Windows-Firewall über die Befehlszeile verwalten:
  https://learn.microsoft.com/de-de/windows/security/operating-system-security/network-security/windows-firewall/configure-with-command-line
- Microsoft Learn – Get-NetFirewallProfile:
  https://learn.microsoft.com/powershell/module/netsecurity/get-netfirewallprofile
- Microsoft Learn – Get-NetFirewallRule:
  https://learn.microsoft.com/powershell/module/netsecurity/get-netfirewallrule
- Microsoft Learn – Get-NetFirewallPortFilter:
  https://learn.microsoft.com/powershell/module/netsecurity/get-netfirewallportfilter
- Microsoft Learn – Get-NetFirewallAddressFilter:
  https://learn.microsoft.com/powershell/module/netsecurity/get-netfirewalladdressfilter
- firewalld – Offizielle Dokumentation:
  https://firewalld.org/documentation/
- firewalld – Handbuch zu `firewall-cmd`:
  https://firewalld.org/documentation/man-pages/firewall-cmd
- Netfilter – nftables Wiki:
  https://wiki.nftables.org/
- Netfilter – nftables-Regelwerk nachverfolgen:
  https://wiki.nftables.org/wiki-nftables/index.php/Ruleset_debug/tracing
- Ubuntu – UFW-Dokumentation:
  https://documentation.ubuntu.com/server/how-to/security/firewalls/
- Linux-Handbuch – `iptables(8)`:
  https://man7.org/linux/man-pages/man8/iptables.8.html
- Linux-Handbuch – `nft(8)`:
  https://man7.org/linux/man-pages/man8/nft.8.html
- Apple – Firewall-Sicherheit bei macOS:
  https://support.apple.com/de-de/guide/security/seca0e83763f/web
- Apple – Verbindungen auf dem Mac mit einer Firewall blockieren:
  https://support.apple.com/de-de/guide/mac-help/mh34041/mac
- Wireshark – Benutzerhandbuch:
  https://www.wireshark.org/docs/wsug_html_chunked/