Klausurvorbereitung 27. Mai

- Zusammenfassung: Schicht 4, Ports, NAT, Firewall und DMZ
- 50 Fragen und Antworten: Schicht 4, Ports, NAT, Firewall und DMZ
- Portforwarding Trainer
- DMZ Firewall Trainer
- NAT(PAT)-Trainer / Source NAT
- TCP-Verbindungsaufbau und TCP-Verbindungsabbau – Trainer

Zusammenfassung: Schicht 4, Ports, NAT, Firewall und DMZ

Zusammenfassung: Schicht 4, Ports, NAT, Firewall und DMZ

Bereich: Seite 64 bis einschließlich Seite 94
Themen: Ports, TCP, UDP, QUIC, Portknocking, Portforwarding, NAT/PAT, Allowlist/Blocklist, Firewall, iptables, DMZ

---

1. Schicht 4 – Transportschicht

Die Schicht 4 ist die Transportschicht. Sie sorgt dafür, dass Daten nicht nur zu einem bestimmten Rechner gelangen, sondern auch zur richtigen Anwendung oder zum richtigen Dienst auf diesem Rechner.

Wichtige Themen der Schicht 4:

- Ports
- TCP
- UDP
- QUIC
- Portknocking
- Portforwarding / Destination NAT
- NAT / PAT / Source NAT
- Allowlist und Blocklist

Die IP-Adresse bestimmt den Host.
Der Port bestimmt den Dienst oder das Programm auf diesem Host.

Beispiel:

```
192.168.1.11:80
```

Das bedeutet:

```
Host: 192.168.1.11
Dienst/Port: 80
```

Port 80 steht typischerweise für HTTP.

Die Kombination aus IP-Adresse und Port nennt man Socket.

Grafik 1: IP-Adresse, Port und Socket

Socket = IP-Adresse + Port IP-Adresse 192.168.1.11 bestimmt den Host
:

Port 80 bestimmt den Dienst Dienst HTTP Webserver

---

2. Warum braucht man Ports?

Ein Rechner kann mehrere Netzwerkprogramme gleichzeitig ausführen.

Beispiele:

- Webbrowser
- Mailprogramm
- Dateifreigabe
- Druckdienst
- Datenbankdienst
- Webserver

Nur mit der IP-Adresse weiß man zwar, welcher Rechner gemeint ist.
Man weiß aber noch nicht, welche Anwendung auf diesem Rechner gemeint ist.

Vergleich:

```
IP-Adresse = Adresse eines Mehrfamilienhauses
Port       = Name oder Wohnung der Person im Haus
```

Beispiel Printserver:

```
10.1.1.1:9100 = Laserdrucker
10.1.1.1:9101 = Tintenstrahldrucker
10.1.1.1:9102 = Nadeldrucker
```

Alle Drucker haben dieselbe IP-Adresse, aber unterschiedliche Ports.

---

3. Schreibweise von IP-Adresse und Port

Bei IPv4 schreibt man:

```
IPv4-Adresse:Port
192.168.1.11:80
```

Bei IPv6 muss die Adresse in eckige Klammern:

```
[IPv6-Adresse]:Port
[2001:1234:5678:90AB:CDEF:1A2B:3C4D:5E6F]:80
```

Wichtig:

```
IP-Adresse + Port = Socket
```

---

4. Aufteilung der Ports

Ports sind in drei große Bereiche eingeteilt.

Bereich	Portnummern	Bedeutung
System Ports	0 bis 1023	feste, bekannte Dienste
User Ports	1024 bis 49151	registrierte Anwendungsports
Dynamic/Private Ports	49152 bis 65535	temporäre Ports für kurzfristige Verbindungen

---

5. Wichtige System Ports

Diese Ports sollte man für Prüfung und Praxis kennen:

Port	Dienst	Protokoll
20 / 21	FTP	TCP
22	SSH	TCP
25	SMTP	TCP
53	DNS	meist UDP
67 / 68	DHCP	UDP
80	HTTP	TCP
110	POP3	TCP
123	NTP	meist UDP
143	IMAP	TCP

Port	Dienst	Protokoll
443	HTTPS	TCP
445	SMB / Samba	TCP

---

6. Wichtige User Ports

Port	Dienst	Protokoll
3128	Squid Proxy	TCP
3306	MySQL	TCP
9100	HP-Druckerport	TCP
10000	Webmin	TCP

---

7. UDP

UDP ist ein einfaches Transportprotokoll.

Eigenschaften:

- verbindungslos
- schneller als TCP
- weniger Verwaltungsaufwand
- keine eingebaute Garantie, dass Pakete ankommen
- keine eingebaute Reihenfolgekontrolle
- wird häufig genutzt, wenn Geschwindigkeit wichtiger ist als perfekte Kontrolle

Typische Beispiele:

- DNS
- DHCP
- NTP
- Streaming
- VoIP
- Online-Gaming

Merksatz:

> UDP ist schnell, aber nicht besonders kontrollierend.

---

8. TCP

TCP ist verbindungsorientiert.

Eigenschaften:

- baut eine Verbindung auf
- bestätigt empfangene Daten
- kontrolliert Reihenfolge und Vollständigkeit
- ist zuverlässiger als UDP
- erzeugt aber mehr Verwaltungsaufwand

Typische Beispiele:

- HTTP
- HTTPS
- SSH
- FTP
- SMTP
- IMAP
- POP3
- SMB

Merksatz:

> TCP ist kontrollierter, aber aufwendiger.

---

9. TCP-Verbindungsaufbau: 3-Wege-Handshake

Beim TCP-Verbindungsaufbau werden drei Schritte genutzt.

1. Client sendet SYN.
2. Server antwortet mit SYN + ACK.
3. Client bestätigt mit ACK.

Danach ist die Verbindung aufgebaut.

Grafik 2: TCP 3-Wege-Handshake

---

10. TCP-Verbindungsabbau: 4-Wege-Handshake

Beim Verbindungsabbau werden vier Schritte genutzt.

1. Client sendet FIN.
2. Server bestätigt mit ACK.
3. Server sendet ebenfalls FIN.
4. Client bestätigt mit ACK.

Danach ist die Verbindung sauber beendet.

Merksatz:

> TCP-Aufbau: 3 Schritte
>
> TCP-Abbau: 4 Schritte

---

11. TCP und Sicherheit

Beim TCP-Verbindungsaufbau kann es zu Angriffen kommen, zum Beispiel SYN-Flood-Angriffen.

Dabei werden viele SYN-Anfragen an einen Server gesendet.
Der Server wartet auf die abschließende Bestätigung, bekommt diese aber nicht oder nicht vollständig.
Dadurch können Ressourcen blockiert werden.

Das gehört zum Bereich Denial-of-Service beziehungsweise Distributed-Denial-of-Service.

---

12. QUIC

QUIC soll Vorteile von UDP und TCP kombinieren.

Eigenschaften:

- basiert technisch auf UDP
- soll schneller und moderner sein
- arbeitet immer mit Verschlüsselung
- wird besonders im modernen Web wichtig

Merksatz:

> QUIC versucht, UDP-Geschwindigkeit mit TCP-ähnlicher Zuverlässigkeit und Verschlüsselung zu verbinden.

---

13. Portknocking

Portknocking bedeutet sinngemäß:

```
Erst richtig anklopfen, dann wird etwas freigegeben.
```

Dabei werden von außen bestimmte Ports in einer festgelegten Reihenfolge angesprochen.

Beispiel:

```
11.1.2.4:1111
11.1.2.4:2222
11.1.2.4:3333
```

Nur wenn diese Reihenfolge stimmt, löst die Firewall oder der Router ein Ereignis aus.

Beispiel aus dem Kochbuch:

```
Portknocking löst Wake-on-LAN aus.
Der interne Webserver startet.
Danach kann ein Zugriff erfolgen.
```

Vorteil:

- Dienste müssen nicht dauerhaft offen sichtbar sein.

Nachteil:

- zusätzlicher Verwaltungsaufwand
- kein Ersatz für echte Authentifizierung oder sichere Dienste

---

14. Portforwarding / Destination NAT

Portforwarding leitet Anfragen von außen nach innen weiter.

Ziel:

```
Internet -> Router/Firewall -> interner Server
```

Beispiel:

```
Externe Adresse: 11.1.2.4:80
Weiterleitung auf: 192.168.178.11:80
```

Oder:

```
11.1.2.4:81 -> 192.168.178.22:80
```

Dadurch können zwei interne Webserver von außen erreichbar gemacht werden, obwohl beide intern Port 80 nutzen.

Wichtig:

Die externe IP-Adresse bleibt gleich.
Die externen Ports unterscheiden, welcher interne Dienst gemeint ist.

Grafik 3: Portforwarding / Destination NAT

Portforwarding / Destination NAT Internet Laptop extern Router / Firewall 11.1.2.4 Port 80 -> Server 1 Port 81 -> Server 2 Webserver 1 192.168.178.11:80 Webserver 2 192.168.178.22:80 11.1.2.4:80 -> :80 -> :80

---

15. Sicherheitsproblem bei Portforwarding

Portforwarding ist praktisch, aber nicht besonders sicher.

Problem:

```
Von außen wird ein direkter Weg in das interne Netz geschaffen.
```

Das kann gefährlich sein, wenn der interne Dienst schlecht abgesichert ist.

Besser:

- möglichst kein unnötiges Portforwarding
- VPN oder Tailscale verwenden
- Dienste aktuell halten
- starke Authentifizierung nutzen
- Firewall-Regeln sauber setzen
- Zugriff einschränken
- Portforwarding eventuell mit Portknocking kombinieren

---

16. NAT / PAT / Source NAT

Im Alltag sagt man oft NAT.
Genauer ist bei vielen Heimroutern eigentlich PAT.

PAT bedeutet Port Address Translation.

Ziel:

```
Mehrere interne Geräte teilen sich eine öffentliche IP-Adresse.
```

Beispiel:

```
Interner PC: 192.168.178.11
Fritz!Box extern: 11.1.2.4
Ziel im Internet: 193.99.144.85:80
```

Der interne PC kann nicht direkt mit seiner privaten IP-Adresse ins Internet.
Die Fritz!Box ersetzt deshalb die interne Quelladresse durch ihre öffentliche Adresse und merkt sich den Zusammenhang in einer Tabelle.

---

17. Ablauf bei NAT / PAT

Beispiel:

```
PC möchte Webseite öffnen:
192.168.178.11:55555 -> 193.99.144.85:80
```

Die Fritz!Box erstellt einen Fake-Port:

```
11.1.2.4:60000 -> 193.99.144.85:80
```

Wenn die Antwort aus dem Internet zurückkommt, schaut die Fritz!Box in ihrer Tabelle nach:

```
11.1.2.4:60000 gehört intern zu 192.168.178.11:55555
```

Dann leitet sie die Antwort an den richtigen internen PC weiter.

Grafik 4: NAT / PAT / Source NAT

NAT / PAT / Source NAT Interner PC 192.168.178.11:55555 Fritz!Box extern: 11.1.2.4 Fake-Port: 60000 merkt sich Zuordnung Webserver 193.99.144.85:80 Anfrage 11.1.2.4:60000 Antwort an Fake-Port zurück an PC

---

18. Unterschied Portforwarding und NAT/PAT

Begriff	Richtung	Zweck
Portforwarding / Destination NAT	extern nach intern	Zugriff aus dem Internet auf internen Dienst
NAT / PAT / Source NAT	intern nach extern	interne Geräte nutzen gemeinsam eine öffentliche IP

Merksatz:

```
Portforwarding: Internet möchte nach innen.
NAT/PAT: internes Netz möchte nach außen.
```

---

19. Allowlist und Blocklist

Früher sagte man Whitelist und Blacklist.
Heute sagt man besser Allowlist und Blocklist.

---

20. Allowlist

Grundprinzip:

```
Alles ist verboten, außer es steht ausdrücklich in der Allowlist.
```

Beispiel:

```
Allowlist:
web.de
```

Dann gilt:

```
web.de   erlaubt
gmx.de   verboten
gmail.de verboten
```

Vorteil:

- sehr streng
- schützt gut vor unerwünschtem Zugriff

Nachteil:

- hoher Pflegeaufwand
- neue erlaubte Seiten müssen ständig ergänzt werden

---

21. Blocklist

Grundprinzip:

```
Alles ist erlaubt, außer es steht ausdrücklich in der Blocklist.
```

Beispiel:

```
Blocklist:
gmx.de
gmail.de
```

Dann gilt:

```
web.de   erlaubt
gmx.de   verboten
gmail.de verboten
```

Vorteil:

- einfacher zu betreiben als reine Allowlist
- gut für bekannte unerwünschte Seiten

Nachteil:

- niemals vollständig
- neue gefährliche Seiten können noch fehlen
- manchmal werden Seiten gesperrt, die eigentlich erlaubt sein sollen

---

22. Kombination aus Allowlist und Blocklist

Praktische Lösung:

```
Zuerst Allowlist prüfen.
Danach Blocklist prüfen.
```

Beispiel:

```
Allowlist:
gmail.de

Blocklist:
gmx.de
gmail.de
```

Dann gilt:

```
web.de   erlaubt, weil in keiner Liste verboten
gmx.de   verboten, weil in Blocklist
gmail.de erlaubt, weil Allowlist Vorrang hat
```

Das ist oft sinnvoller als nur eine reine Allowlist oder nur eine reine Blocklist.

---

23. SquidGuard

SquidGuard kann Webseiten nach Kategorien filtern.

Beispiele für Kategorien:

- Dating
- Mailing
- Hacking
- Werbung
- Malware
- Glücksspiel

Das Ausrufezeichen bedeutet „nicht“ beziehungsweise „verboten“.

Beispielhafte Logik:

```
Allowlist !Dating !Mailing !Hacking any
```

Sinngemäß:

1. Prüfe zuerst Allowlist.
2. Wenn Treffer in Allowlist: erlauben.
3. Wenn kein Treffer: prüfe verbotene Kategorien.
4. Wenn Treffer in verbotener Kategorie: sperren.

5. Wenn kein Treffer: erlauben.

---

24. Firewalls

Eine Firewall kontrolliert Netzwerkverkehr anhand von Regeln.

Sie entscheidet:

```
Darf dieses Paket durch?
Ja oder Nein?
```

Firewalls können auf verschiedenen Ebenen arbeiten:

- Host-Firewall
- Unternehmens-Firewall
- Paketfilter-Firewall
- Stateful-Packet-Inspection-Firewall

---

25. Personal-Firewall und Unternehmens-Firewall

Firewall-Typ	Aufgabe
Personal-Firewall	schützt einen einzelnen PC oder Server
Unternehmens-Firewall	schützt ein ganzes LAN oder Teilnetze

---

26. Paketfilter-Firewall

Eine Paketfilter-Firewall ist einfacher und älter.

Problem:

Bei klassischen Paketfiltern muss häufig Hinweg und Rückweg erlaubt werden.

Beispiel:

```
Client -> Server erlauben
Server -> Client ebenfalls erlauben
```

Das ist fehleranfälliger.

---

27. Stateful Packet Inspection Firewall

Eine SPI-Firewall merkt sich den Zustand einer Verbindung.

Vorteil:

> Nur der Hinweg muss ausdrücklich erlaubt werden.
> Der Rückweg wird automatisch als passende Antwort erkannt.

Das ist moderner und sicherer.

Beispiel:

> Intern -> Internet erlaubt
> Antwort Internet -> Intern wird automatisch erkannt

---

28. Steuerbare Schichten einer SPI-Firewall

Eine SPI-Firewall kann mehrere Bedingungen prüfen.

OSI-Schicht	Prüfbares Merkmal
Schicht 2	MAC-Adresse
Schicht 3	IP-Adresse
Schicht 4	TCP oder UDP
Schicht 5 bis 7	Port / Anwendung

Wichtig:

> Alle angegebenen Bedingungen sind logisch UND-verknüpft.

Beispiel:

> Nur diese MAC-Adresse
> UND nur diese IP-Adresse
> UND nur TCP
> UND nur Port 22

Dann darf nur genau dieser passende Verkehr durch.

Wichtiger Merksatz:

Was nicht abgefragt wird, ist erlaubt.

Das bedeutet:

Wenn eine Regel keine MAC-Adresse prüft, ist die MAC-Adresse für diese Regel egal.

---

29. Firewall als Brücke zwischen intern und extern

Im Kochbuch wird die Firewall mit einer Brücke zwischen Insel und Festland verglichen.

- Insel = internes sicheres Netz
- Festland = externes unsicheres Netz
- Brücke = Firewall
- Wachmannschaft = Linux-System mit Firewall
- grüne Seite = internes Netz
- rote Seite = externes Netz

Die Firewall kontrolliert, wer von innen nach außen und von außen nach innen darf.

---

30. iptables: INPUT, OUTPUT und FORWARD

Bei iptables gibt es drei wichtige Regelketten.

Regelkette	Bedeutung
INPUT	Verkehr zur Firewall selbst
OUTPUT	Verkehr von der Firewall selbst nach außen
FORWARD	Verkehr, der durch die Firewall hindurchgeleitet wird

---

Grafik 5: INPUT, OUTPUT und FORWARD

iptables: INPUT, OUTPUT und FORWARD Intern eth0 / grün Firewall Linux INPUT OUTPUT FORWARD Extern eth1 / rot INPUT OUTPUT FORWARD = Verkehr durch die Firewall hindurch

---

31. FORWARD

FORWARD betrifft Datenverkehr, der durch die Firewall hindurchgeht.

Beispiele:

```
internes Netz -> Firewall -> Internet
Internet -> Firewall -> internes Netz
```

FORWARD ist wichtig, wenn die Firewall als Router zwischen zwei Netzen arbeitet.

---

32. INPUT

INPUT betrifft Datenverkehr, der direkt an die Firewall selbst gerichtet ist.

Beispiele:

```
Admin-PC -> Firewall per SSH
Monitoring-Server -> Firewall
Ping an Firewall
```

---

33. OUTPUT

OUTPUT betrifft Datenverkehr, der von der Firewall selbst erzeugt wird.

Beispiele:

```
Firewall -> NTP-Server
Firewall -> DNS-Server
Firewall -> Update-Server
```

---

34. Beispiel einfache Firewall

Beim Erstellen einer Firewall-Regel muss man sich immer fragen:

- Wo ist die Quelle?
- Wo ist das Ziel?
- Ist es INPUT, OUTPUT oder FORWARD?
- Wird TCP, UDP oder ICMP genutzt?
- Welcher Port wird genutzt?
- Ist die IP-Adresse ein einzelner Host oder ein ganzes Netz?
- Muss eine Subnetzmaske angegeben werden?
- Soll zusätzlich die MAC-Adresse geprüft werden?

Wichtig:

```
ICMP hat keinen Port.
```

---

35. iptables-Syntax aus dem Kochbuch

Beispielhafte Regeln:

```
$FT $MP -s 192.168.2.0/24 -d 192.168.1.2 --dports 80,3128 $R
$IT $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.2.1 --dport 22 $R
$OU -s 192.168.1.1 -d 192.168.1.2 --dport 123 $R
```

Bedeutung:

Kürzel / Option	Bedeutung
FT	FORWARD TCP
IT	INPUT TCP
OU	OUTPUT UDP
MP	Multiport, also mehrere Ports
-s	Source / Quelle
-d	Destination / Ziel
--dport	ein Ziel-Port
--dports	mehrere Ziel-Ports
MAC	MAC-Adresse zusätzlich prüfen
ACCEPT	Regel erlaubt den Verkehr
DROP	Paket wird verworfen

---

36. Catch-all-Regel

Am Ende steht häufig:

```
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
iptables -A FORWARD -j DROP
```

Bedeutung:

```
Alles, was vorher nicht ausdrücklich erlaubt wurde, wird verboten.
```

Das ist ein sehr wichtiges Firewall-Prinzip.

Merksatz:

```
Erst erlauben, was gebraucht wird.
Dann alles andere blockieren.
```

---

37. Praktische Umsetzung einer einfachen Firewall

Für das einfache Beispiel braucht man:

- Linux-PC
- zwei Netzwerkkarten
- Root-Rechte
- iptables-Skript
- passende IP-Adressen auf den Netzwerkkarten

Beispiel:

```
Netzwerkkarte 1: 192.168.1.1
Netzwerkkarte 2: 192.168.2.1
```

Typische Befehle:

```
sudo -s
chmod 755 alle.sh
./alle.sh
```

---

38. DMZ – Demilitarisierte Zone

Eine DMZ ist ein separates Zwischennetz.

Sie wird genutzt für Server, die sowohl von intern als auch von extern erreichbar sein sollen.

Beispiele:

- Webserver
- Mailserver

- DNS-Server
- Reverse Proxy

Die DMZ liegt nicht direkt im internen LAN.

Ziel:

```
Wenn ein öffentlicher Server angegriffen wird, soll nicht direkt das interne LAN betroffen sein.
```

---

39. DMZ-Farben

Typische Darstellung:

Bereich	Farbe
Internes LAN	grün
Externes Netz / Internet	rot
DMZ	orange

---

40. Zwei-stufiges DMZ-Konzept

Das zwei-stufige Konzept nutzt zwei Firewalls.

Aufbau:

```
Internet -> Firewall 1 -> DMZ -> Firewall 2 -> internes LAN
```

Vorteile:

- sicherer
- doppelte Verteidigungslinie
- in IHK-Prüfungen oft bevorzugt

Nachteile:

- aufwendiger
- zwei Firewalls
- zwei Regelwerke
- Routing muss sauber geplant werden

Merksatz:

```
Zwei-stufige DMZ = sicherer, aber aufwendiger.
```

---

41. Ein-stufiges DMZ-Konzept

Das ein-stufige Konzept nutzt eine Firewall mit drei Netzwerkkarten.

Aufbau:

```
Internes LAN
    |
Firewall mit 3 Netzwerkkarten
    |
Internet
    |
DMZ
```

Vorteile:

- weniger Aufwand
- nur eine Firewall
- nur ein Regelwerk

Nachteile:

- weniger sicher als zwei-stufiges Konzept
- nur eine Verteidigungslinie

Merksatz:

```
Ein-stufige DMZ = einfacher, aber unsicherer.
```

Grafik 6: Ein-stufige DMZ mit drei Netzwerkkarten

Ein-stufige DMZ Firewall 3 Netzwerkkarten Internes LAN grün Internet rot DMZ orange

---

Grafik 7: Zwei-stufige DMZ

Zwei-stufige DMZ Internet rot Firewall 1 DMZ orange Firewall 2 Internes LAN

---

42. Komplexere Firewall mit DMZ

Im komplexeren Beispiel wird eine Linux-Maschine mit drei Netzwerkkarten als Firewall eingesetzt.

Beispielhafte Netze:

```
Externes Netz: 192.168.1.0/24
Internes Netz: 192.168.2.0/24
DMZ:           192.168.3.0/24
```

Die Firewall hat dann zum Beispiel:

```
1. Netzwerkkarte: 192.168.1.1
2. Netzwerkkarte: 192.168.2.1
3. Netzwerkkarte: 192.168.3.1
```

Regeln können dann zum Beispiel erlauben:

- internes Netz darf Web/Proxy nach extern nutzen
- interner Host darf per SSH auf Firewall
- interner Admin-PC darf per SSH auf DMZ-Server
- internes Netz darf DNS-Server in der DMZ nutzen
- Firewall darf NTP nach außen nutzen

Alles andere wird durch DROP verworfen.

---

43. Prüfungssichere Merksätze

```
IP-Adresse = welcher Host?
Port = welcher Dienst?
Socket = IP-Adresse + Port

TCP (Transmission Control Protocol) = verbindungsorientiert und zuverlässig

TCP ist wie ein Einschreiben mit Rückschein. Es ist extrem zuverlässig, aber durch die vielen Kontrollen etwas langsamer.
TCP baut vor der eigentlichen Datenübertragung eine Verbindung auf. Dafür wird der sogenannte Three-Way-Handshake verwendet.

Vereinfacht:

1. Client fragt: Darf ich eine Verbindung aufbauen?
```

2. Server antwortet: Ja, ich bin bereit.
3. Client bestätigt: Verbindung steht.

UDP (User Datagram Protocol) = verbindungslos und schnell
Verbindungslos: Es wird keine formelle Verbindung (kein "Handshake") zwischen Sender und Empfänger aufgebaut,
bevor Daten gesendet werden. Daten werden einfach "abgeschickt"

Unzuverlässig: Es gibt keine Empfangsbestätigung und keine Prüfung, ob ein Paket angekommen ist.
Geht ein Paket verloren, wird es nicht erneut gesendet.

UDP ist wie das Werfen eines Balls. Es sendet Daten einfach los, ohne zu prüfen, ob sie ankommen.
Es ist rasend schnell und wird daher oft für Live-Streaming, Online-Gaming oder Telefonie (VoIP) genutzt.

QUIC = modernes Protokoll auf UDP-Basis mit Verschlüsselung
Portforwarding = von außen nach innen
NAT/PAT (Network/Port Address Translation) = von innen nach außen über eine gemeinsame öffentliche IP
NAT/PAT = interne private Adressen werden beim Zugriff nach außen in eine öffentliche Adresse übersetzt.

Allowlist = alles verboten außer erlaubt
Blocklist = alles erlaubt außer verboten

SPI-Firewall (Stateful Packet Inspection) = merkt sich Verbindungszustände
Erklärung: ist eine hochentwickelte Sicherheitstechnologie in Routern und Firewalls. Sie schützt Netzwerke, indem sie Datenpakete nicht nur einzeln bewertet, sondern den gesamten Kontext und Verbindungsstatus (den "Zustand") einer Kommunikation überwacht und speichert.

INPUT = zur Firewall selbst
OUTPUT = von der Firewall selbst
FORWARD = durch die Firewall hindurch
Portforwarding = eingehende Verbindung von außen wird gezielt an ein internes Gerät weitergeleitet.
DMZ = separates Zwischennetz für öffentlich erreichbare Server
Zwei-stufige DMZ = sicherer, aber aufwendiger
Ein-stufige DMZ = einfacher, aber unsicherer

---

44. Mini-Vergleich: wichtigste Begriffe

Begriff	Kurz erklärt
Port	Nummer für Dienst oder Anwendung

Begriff	Kurz erklärt
Socket	Kombination aus IP-Adresse und Port
TCP	zuverlässige Verbindung mit Aufbau und Abbau
UDP	schnelle, verbindungslose Übertragung
QUIC	modernes, verschlüsseltes Protokoll auf UDP-Basis
Portknocking	richtige Port-Reihenfolge löst Ereignis aus
Portforwarding	externer Port wird auf internen Dienst weitergeleitet
NAT/PAT	interne Geräte teilen sich öffentliche IP-Adresse
Allowlist	nur ausdrücklich Erlaubtes ist erlaubt
Blocklist	alles erlaubt außer ausdrücklich Verbotenem
Firewall	kontrolliert Netzwerkverkehr anhand von Regeln
SPI	Stateful Packet Inspection, merkt sich Verbindungen
INPUT	Pakete zur Firewall
OUTPUT	Pakete von der Firewall
FORWARD	Pakete durch die Firewall
DMZ	separates Netz für Dienste zwischen intern und extern

---

45. Typische Prüfungsfrage: Was ist der Unterschied zwischen NAT und Portforwarding?

NAT/PAT wird genutzt, wenn interne Geräte ins Internet wollen.
Dabei ersetzt der Router die private Quelladresse durch seine öffentliche IP-Adresse und merkt sich die Verbindung über Ports.

Portforwarding wird genutzt, wenn externe Geräte aus dem Internet auf einen internen Dienst zugreifen sollen.
Dabei wird ein externer Port auf eine interne IP-Adresse und einen internen Port weitergeleitet.

Kurz:

```
NAT/PAT: innen -> außen
Portforwarding: außen -> innen
```

---

46. Typische Prüfungsfrage: Warum ist eine DMZ sinnvoll?

Eine DMZ trennt öffentlich erreichbare Server vom internen LAN.

Wenn ein Webserver in der DMZ angegriffen oder kompromittiert wird, liegt er nicht direkt im internen Netz.
Dadurch wird das interne LAN besser geschützt.

Kurz:

```
DMZ = Sicherheitszone zwischen Internet und internem LAN.
```

---

47. Typische Prüfungsfrage: Warum ist eine SPI-Firewall besser als ein einfacher Paketfilter?

Eine SPI-Firewall merkt sich den Zustand einer Verbindung.

Wenn ein interner Client eine Verbindung nach außen aufbaut, erkennt die Firewall die passende Antwort automatisch.
Man muss den Rückweg nicht separat freigeben.

Kurz:

```
Paketfilter: Hinweg und Rückweg oft manuell regeln.
SPI-Firewall: Hinweg erlauben, Rückweg wird passend erkannt.
```

---

48. Typische Prüfungsfrage: Was bedeutet „Default DROP“?

Default DROP bedeutet:

```
Alles, was nicht ausdrücklich erlaubt ist, wird verworfen.
```

Das ist ein sicheres Firewall-Grundprinzip.

Man erstellt zuerst die notwendigen Erlaubnisregeln.
Am Ende wird alles andere blockiert.

---

49. Gesamtbild

Die Seiten 64 bis 94 zeigen den Übergang von der Transportschicht zur praktischen Netzwerksicherheit.

Erst wird erklärt, wie Dienste über Ports unterschieden werden.
Danach wird gezeigt, wie TCP, UDP und QUIC arbeiten.
Anschließend geht es darum, wie Verbindungen nach innen oder außen weitergeleitet werden.

Zum Schluss werden Firewalls, iptables-Regeln und DMZ-Konzepte erklärt.

Das zentrale Prinzip lautet:

> Netzwerkkommunikation muss adressiert, unterschieden, kontrolliert und abgesichert werden.

50 Fragen und Antworten: Schicht 4, Ports, NAT, Firewall und DMZ

50 Fragen und Antworten: Schicht 4, Ports, NAT, Firewall und DMZ

---

1. Was ist die Aufgabe der Schicht 4 im Netzwerkmodell?

Die Schicht 4 ist die Transportschicht. Sie sorgt dafür, dass Daten nicht nur beim richtigen Rechner ankommen, sondern auch bei der richtigen Anwendung oder dem richtigen Dienst auf diesem Rechner.

---

2. Warum reicht eine IP-Adresse allein nicht aus?

Eine IP-Adresse bestimmt nur den Zielrechner. Auf einem Rechner können aber mehrere Dienste gleichzeitig laufen, zum Beispiel Webserver, Maildienst oder Dateifreigabe. Deshalb braucht man zusätzlich einen Port, um den richtigen Dienst anzusprechen.

---

3. Was ist ein Port?

Ein Port ist eine Nummer, über die ein bestimmter Dienst oder eine Anwendung auf einem Rechner angesprochen wird.

Beispiel:

```
192.168.1.11:80
```

Die IP-Adresse ist `192.168.1.11`.
Der Port ist `80`.
Port 80 steht typischerweise für HTTP.

---

4. Was ist ein Socket?

Ein Socket ist die Kombination aus IP-Adresse und Port.

```
Socket = IP-Adresse + Port
```

Beispiel:

```
192.168.1.11:80
```

---

5. Was bedeutet die Schreibweise 192.168.1.11:80?

Diese Schreibweise bedeutet:

```
192.168.1.11 = Zielhost
80           = Zielport
```

Der Zugriff geht also an den Rechner mit der IP-Adresse `192.168.1.11` und dort an den Dienst auf Port `80`.

---

6. Wie schreibt man eine IPv6-Adresse mit Port korrekt?

Bei IPv6 muss die Adresse in eckige Klammern gesetzt werden, damit der Port eindeutig erkennbar ist.

```
[2001:1234:5678:90AB:CDEF:1A2B:3C4D:5E6F]:80
```

Ohne Klammern wäre nicht klar erkennbar, wo die IPv6-Adresse endet und wo der Port beginnt.

---

7. In welche drei Bereiche werden Ports eingeteilt?

Ports werden in drei Bereiche eingeteilt:

```
0 bis 1023      = System Ports
1024 bis 49151  = User Ports
49152 bis 65535 = Dynamic/Private Ports
```

8. Was sind System Ports?

System Ports sind die bekannten Ports von `0` bis `1023`. Sie werden für wichtige Standarddienste verwendet.

Beispiele:

```
22  = SSH
53  = DNS
80  = HTTP
443 = HTTPS
```

9. Was sind User Ports?

User Ports liegen im Bereich von `1024` bis `49151`. Sie werden für registrierte Anwendungen und Dienste verwendet.

Beispiele:

```
3128  = Squid Proxy
3306  = MySQL
9100  = Druckerport
10000 = Webmin
```

10. Was sind Dynamic oder Private Ports?

Dynamic oder Private Ports liegen im Bereich von `49152` bis `65535`.

Sie werden oft automatisch und kurzfristig von Clients verwendet, wenn eine Verbindung aufgebaut wird.

11. Welcher Port wird typischerweise für HTTP verwendet?

HTTP verwendet typischerweise Port `80`.

```
HTTP = Port 80
```

---

12. Welcher Port wird typischerweise für HTTPS verwendet?

HTTPS verwendet typischerweise Port `443`.

```
HTTPS = Port 443
```

---

13. Welcher Port wird typischerweise für SSH verwendet?

SSH verwendet typischerweise Port `22`.

```
SSH = Port 22
```

SSH wird häufig zur sicheren Fernadministration von Linux-Systemen genutzt.

---

14. Welcher Port wird typischerweise für DNS verwendet?

DNS verwendet typischerweise Port `53`.

DNS nutzt meistens UDP, kann aber in bestimmten Fällen auch TCP verwenden.

---

15. Welche Ports verwendet DHCP?

DHCP verwendet die Ports `67` und `68`.

```
Port 67 = DHCP-Server
Port 68 = DHCP-Client
```

DHCP nutzt UDP.

---

16. Was ist UDP?

UDP ist ein verbindungsloses Transportprotokoll.

Eigenschaften:

```
schnell
wenig Verwaltungsaufwand
keine feste Verbindung
keine Garantie für Zustellung
keine eingebaute Reihenfolgekontrolle
```

---

17. Wann wird UDP häufig verwendet?

UDP wird häufig verwendet, wenn Geschwindigkeit wichtiger ist als vollständige Kontrolle.

Beispiele:

```
DNS
DHCP
NTP
Streaming
VoIP
Online-Gaming
```

---

18. Was ist TCP?

TCP ist ein verbindungsorientiertes Transportprotokoll.

Eigenschaften:

```
Verbindungsaufbau
Bestätigung empfangener Daten
Kontrolle der Reihenfolge
zuverlässige Datenübertragung
```

```
mehr Verwaltungsaufwand als UDP
```

19. Wann wird TCP häufig verwendet?

TCP wird genutzt, wenn Daten zuverlässig und vollständig übertragen werden müssen.

Beispiele:

```
HTTP
HTTPS
SSH
FTP
SMTP
IMAP
POP3
SMB
```

20. Was ist der wichtigste Unterschied zwischen TCP und UDP?

TCP ist verbindungsorientiert und kontrolliert die Übertragung.
UDP ist verbindungslos und schneller, aber ohne eingebaute Zustellgarantie.

Kurz:

```
TCP = zuverlässig, aber aufwendiger
UDP = schneller, aber weniger kontrolliert
```

21. Was ist der TCP-3-Wege-Handshake?

Der TCP-3-Wege-Handshake ist der Verbindungsaufbau bei TCP.

Ablauf:

```
1. Client -> Server: SYN
2. Server -> Client: SYN + ACK
3. Client -> Server: ACK
```

Danach ist die TCP-Verbindung aufgebaut.

---

22. Wofür steht SYN?

SYN steht für Synchronisation.

Beim TCP-Verbindungsaufbau sendet der Client zuerst ein SYN-Paket an den Server, um eine Verbindung zu starten.

---

23. Wofür steht ACK?

ACK steht für Acknowledgement, also Bestätigung.

Ein ACK bestätigt, dass ein Paket oder ein Verbindungsschritt angekommen ist.

---

24. Wie wird eine TCP-Verbindung kontrolliert abgebaut?

Der TCP-Verbindungsabbau erfolgt typischerweise in vier Schritten:

```
1. FIN
2. ACK
3. FIN
4. ACK
```

Merksatz:

```
TCP-Aufbau = 3 Schritte
TCP-Abbau  = 4 Schritte
```

---

25. Was ist ein SYN-Flood-Angriff?

Bei einem SYN-Flood-Angriff werden sehr viele SYN-Anfragen an einen Server gesendet.

Der Server reserviert Ressourcen für halboffene Verbindungen. Wenn die abschließende Bestätigung ausbleibt, können diese Ressourcen blockiert werden.

Das kann zu einer Überlastung des Servers führen.

---

26. Was ist QUIC?

QUIC ist ein modernes Transportprotokoll, das auf UDP basiert.

Es soll Geschwindigkeit, Zuverlässigkeit und Verschlüsselung verbinden.

Kurz:

```
QUIC = UDP-Basis + moderne Verbindungskontrolle + Verschlüsselung
```

---

27. Warum ist QUIC im modernen Web wichtig?

QUIC ist wichtig, weil es schnelle Verbindungen ermöglichen soll und gleichzeitig Verschlüsselung fest integriert.

Es wird vor allem für moderne Webkommunikation eingesetzt.

---

28. Was ist Portknocking?

Portknocking bedeutet, dass bestimmte Ports in einer festgelegten Reihenfolge angesprochen werden müssen.

Nur wenn die Reihenfolge stimmt, wird eine Aktion ausgelöst.

Beispiel:

```
Port 1111
Port 2222
Port 3333
```

Erst danach wird ein Dienst freigegeben oder ein Ereignis ausgelöst.

---

29. Was ist der Vorteil von Portknocking?

Der Vorteil ist, dass ein Dienst nicht dauerhaft offen sichtbar sein muss.

Er wird erst nach der richtigen „Anklopf-Reihenfolge“ erreichbar oder aktiviert.

---

30. Ist Portknocking ein Ersatz für sichere Authentifizierung?

Nein.

Portknocking kann eine zusätzliche Schutzmaßnahme sein, ersetzt aber keine sichere Authentifizierung, keine starken Passwörter und keine sauber abgesicherten Dienste.

---

31. Was ist Portforwarding?

Portforwarding leitet eine Anfrage von außen an einen internen Dienst weiter.

Beispiel:

```
11.1.2.4:80 -> 192.168.178.11:80
```

Eine externe Anfrage an Port `80` der öffentlichen Adresse wird an den internen Webserver weitergeleitet.

---

32. Was ist Destination NAT?

Destination NAT bedeutet, dass die Zieladresse oder der Zielport eines Pakets verändert wird.

Beim Portforwarding wird zum Beispiel aus:

```
11.1.2.4:80
```

intern:

```
192.168.178.11:80
```

---

33. Warum kann Portforwarding gefährlich sein?

Portforwarding öffnet einen Weg aus dem Internet in das interne Netzwerk.

Wenn der interne Dienst schlecht abgesichert oder veraltet ist, kann das ein Sicherheitsrisiko darstellen.

---

34. Welche sichere Alternative gibt es oft zu Portforwarding?

Eine sichere Alternative ist häufig ein VPN oder ein privater Zugriffsdienst wie Tailscale.

Dann muss ein Dienst nicht direkt öffentlich aus dem Internet erreichbar sein.

---

35. Was ist NAT?

NAT bedeutet Network Address Translation.

Dabei werden IP-Adressen beim Übergang zwischen zwei Netzen umgeschrieben.

Im Heimnetz wird meist die private interne IP-Adresse durch die öffentliche IP-Adresse des Routers ersetzt.

---

36. Was ist PAT?

PAT bedeutet Port Address Translation.

Dabei teilen sich mehrere interne Geräte eine öffentliche IP-Adresse. Der Router unterscheidet die Verbindungen über Ports.

Kurz:

```
Viele interne Geräte -> eine öffentliche IP
Unterscheidung über Ports
```

---

37. Was ist Source NAT?

Source NAT bedeutet, dass die Quelladresse eines Pakets verändert wird.

Beispiel:

```
Interner PC: 192.168.178.11
Router extern: 11.1.2.4
```

Wenn der PC ins Internet geht, ersetzt der Router die private Quelladresse durch seine öffentliche Adresse.

---

38. Was ist der Unterschied zwischen NAT/PAT und Portforwarding?

NAT/PAT wird verwendet, wenn interne Geräte nach außen ins Internet kommunizieren.

Portforwarding wird verwendet, wenn externe Geräte von außen auf einen internen Dienst zugreifen sollen.

Kurz:

```
NAT/PAT       = innen nach außen
Portforwarding = außen nach innen
```

---

39. Was ist eine Allowlist?

Eine Allowlist arbeitet nach dem Prinzip:

```
Alles ist verboten, außer es ist ausdrücklich erlaubt.
```

Nur Einträge, die auf der Allowlist stehen, dürfen genutzt werden.

---

40. Was ist eine Blocklist?

Eine Blocklist arbeitet nach dem Prinzip:

```
Alles ist erlaubt, außer es ist ausdrücklich verboten.
```

Nur Einträge, die auf der Blocklist stehen, werden gesperrt.

---

41. Was ist strenger: Allowlist oder Blocklist?

Eine Allowlist ist strenger.

Bei einer Allowlist ist zunächst alles verboten. Nur ausdrücklich erlaubte Ziele oder Dienste dürfen genutzt werden.

---

42. Was ist der Nachteil einer reinen Allowlist?

Eine reine Allowlist hat einen hohen Pflegeaufwand.

Jede erlaubte Webseite, jeder erlaubte Dienst oder jedes erlaubte Ziel muss vorher eingetragen werden.

---

43. Was ist der Nachteil einer reinen Blocklist?

Eine Blocklist ist nie vollständig.

Neue gefährliche oder unerwünschte Seiten können fehlen und dadurch trotzdem erreichbar sein.

44. Was ist SquidGuard?

SquidGuard ist ein Filterwerkzeug, das Webseiten anhand von Listen oder Kategorien blockieren oder erlauben kann.

Beispiele für Kategorien:

```
Dating
Mailing
Hacking
Werbung
Malware
Glücksspiel
```

45. Was ist eine Firewall?

Eine Firewall kontrolliert Netzwerkverkehr anhand von Regeln.

Sie entscheidet, ob ein Paket erlaubt oder blockiert wird.

Kurz:

```
Firewall = kontrolliert Datenverkehr
```

46. Was ist eine Personal-Firewall?

Eine Personal-Firewall schützt einen einzelnen Rechner.

Sie läuft direkt auf dem PC oder Server und kontrolliert dessen ein- und ausgehenden Netzwerkverkehr.

47. Was ist eine Unternehmens-Firewall?

Eine Unternehmens-Firewall schützt ein ganzes Netzwerk oder mehrere Teilnetze.

Sie steht meistens zwischen internem LAN und externem Netz beziehungsweise Internet.

---

48. Was ist der Unterschied zwischen Paketfilter und Stateful Packet Inspection?

Ein einfacher Paketfilter prüft einzelne Pakete anhand von Regeln.

Eine Stateful-Packet-Inspection-Firewall merkt sich zusätzlich den Zustand einer Verbindung.

Vorteil von SPI:

```
Der Hinweg wird erlaubt.
Der passende Rückweg wird automatisch erkannt.
```

---

49. Was bedeuten INPUT, OUTPUT und FORWARD bei iptables?

Bei iptables gibt es wichtige Regelketten:

```
INPUT   = Verkehr zur Firewall selbst
OUTPUT  = Verkehr von der Firewall selbst nach außen
FORWARD = Verkehr durch die Firewall hindurch
```

Beispiele:

```
INPUT   = SSH-Zugriff auf die Firewall
OUTPUT  = Firewall fragt NTP-Server ab
FORWARD = PC im LAN geht über Firewall ins Internet
```

---

50. Was ist eine DMZ und warum ist sie sinnvoll?

Eine DMZ ist eine demilitarisierte Zone, also ein separates Zwischennetz.

Dort stehen Server, die von außen erreichbar sein müssen, zum Beispiel:

```
Webserver
Mailserver
DNS-Server
Reverse Proxy
```

Der Vorteil:

Wenn ein Server in der DMZ angegriffen wird, befindet er sich nicht direkt im internen LAN.

Kurz:

```
DMZ = Sicherheitszone zwischen Internet und internem Netzwerk
```

Portforwarding Trainer

https://trainer.ulrich-wiki.com/portforwarding-trainer.html?v=9

DMZ Firewall Trainer

DMZ-Firewall-Trainer

https://trainer.ulrich-wiki.com/dmz-firewall-trainer-wide-v9-grafik-final.html?v=1

DMZ-Firewall-Trainer im Vollbild öffnen

Korrektur / wichtige Hinweise

Die externe öffentliche IP-Adresse ist:

```
11.1.2.4
```

Wichtig zu Regel (7):

Bei **Regel (7)** darf **nicht** `11.1.2.4` als Quell- oder Ziel-IP eingetragen werden.

Korrekt ist:

```
Richtung:     OUTPUT
Protokoll:    UDP
Quell-IP:     192.168.1.1
Ziel-IP:      192.168.1.2
Ziel-Port(s): 123
```

`11.1.2.4` ist nur die öffentliche/externe Adresse in der Skizze. Regel (7) beschreibt dagegen:

```
Firewall → Proxy
192.168.1.1 → 192.168.1.2
UDP Port 123
```

Korrektur zu „(weitere)"

Bei **(weitere)** ist hier **ping** gemeint.

```
(weitere)
ping
```

Fachlich gehört ping zu:

```
ICMP
```

Die passende Regel ist sinngemäß:

```
INPUT
ICMP
Quelle: Admin-PC / 192.168.2.2
Ziel: Firewall intern / 192.168.2.1
Ziel-Port: leer, weil ICMP keinen TCP-/UDP-Port verwendet
Kommentar: Admin darf die Firewall anpingen
```

Merksatz

```
FORWARD = Paket läuft durch die Firewall hindurch.
INPUT   = Paket geht an die Firewall selbst.
OUTPUT  = Paket kommt von der Firewall selbst.
```

NAT(PAT)-Trainer / Source NAT

https://trainer.ulrich-wiki.com/nat-pat-trainer.html?v=7

NAT/PAT-Trainer im Vollbild öffnen

TCP-Verbindungsaufbau und TCP-Verbindungsabbau – Trainer

TCP-Verbindungsaufbau und TCP-Verbindungsabbau – Trainer

Dieser interaktive Trainer gehört zu den Aufgaben:

> Frage 2: TCP-Verbindungsaufbau mit Sequenznummern
>
> Frage 3: TCP-Verbindungsabbau mit Sequenznummern

Die Grundlage ist die Klausurübung Netzwerktechnik Schicht 4–7. Dort werden beim TCP-Aufbau und TCP-Abbau die Begriffe `SYN`, `FIN`, `seq`, `ACK`, `x`, `x+1`, `y` und `y+1` verwendet.

Interaktiver TCP-Trainer

https://trainer.ulrich-wiki.com/tcp-verbindung-trainer.html?v=2

TCP-Trainer im Vollbild öffnen

Was wird trainiert?

Bereich	Bedeutung
SYN	Startet den TCP-Verbindungsaufbau
SYN ACK	Server bestätigt SYN und sendet eigene Sequenznummer
ACK	Bestätigung einer Sequenznummer
FIN	Beendet eine Richtung der TCP-Verbindung
seq	Sequenznummer des sendenden Hosts
ACK-Nummer	Bestätigt die nächste erwartete Sequenznummer
x+1 / y+1	SYN und FIN verbrauchen jeweils eine Sequenznummer

Merksatz für den TCP-Verbindungsaufbau

> 1. Client → Server: SYN, seq = x
> 2. Server → Client: SYN + ACK, seq = y, ack = x+1

```
3. Client → Server: ACK, seq = x+1, ack = y+1
```

Beispiel:

```
x = 1000
y = 2000

x+1 = 1001
y+1 = 2001
```

Merksatz für den TCP-Verbindungsabbau

```
1. Client → Server: FIN, seq = x
2. Server → Client: ACK, ack = x+1
3. Server → Client: FIN, seq = y
4. Client → Server: ACK, ack = y+1
```

Beispiel:

```
x = 1000
y = 2000

x+1 = 1001
y+1 = 2001
```

Wichtige Regel

```
SYN verbraucht eine Sequenznummer.
FIN verbraucht eine Sequenznummer.
Deshalb wird jeweils mit +1 bestätigt.
```

Typische Fehler

Fehler	Warum falsch?
SYN ohne ACK im zweiten Schritt	Der Server muss das SYN des Clients bestätigen
ACK-Zahl nicht +1	SYN/FIN wurden nicht korrekt bestätigt

Fehler	Warum falsch?
Pfeilrichtung falsch	Sender und Empfänger werden vertauscht
FIN-Abbau nur mit 3 Schritten	Der normale TCP-Abbau wird meist mit 4 Schritten dargestellt
seq und ACK verwechselt	seq ist die eigene Nummer, ACK bestätigt die Gegenseite

---

Mini-Testfragen

1. Wofür steht SYN?

`SYN` steht für Synchronisation.

Es wird beim TCP-Verbindungsaufbau verwendet, um Sequenznummern zu synchronisieren.

2. Was passiert beim zweiten Schritt des TCP-Verbindungsaufbaus?

Der Server antwortet mit:

```
SYN + ACK
seq = y
ack = x+1
```

Damit bestätigt er das SYN des Clients und sendet seine eigene Sequenznummer.

3. Warum wird aus x die ACK-Nummer x+1?

Weil SYN eine Sequenznummer verbraucht.

Wenn der Client `seq = x` sendet, erwartet der Server danach `x+1`.

4. Wofür steht FIN?

`FIN` steht für Finish.

Es wird verwendet, um eine TCP-Verbindung beziehungsweise eine Kommunikationsrichtung zu beenden.

5. Warum wird FIN ebenfalls mit +1 bestätigt?

Weil FIN ebenfalls eine Sequenznummer verbraucht.

Darum wird ein FIN mit der nächsten erwarteten Nummer bestätigt:

```
ack = x+1
```

6. Wie viele Schritte hat der TCP-Verbindungsaufbau?

Der TCP-Verbindungsaufbau hat 3 Schritte.

```
SYN
SYN + ACK
ACK
```

7. Wie viele Schritte hat der TCP-Verbindungsabbau?

Der TCP-Verbindungsabbau wird typischerweise mit 4 Schritten dargestellt.

```
FIN
ACK
FIN
ACK
```

Nächste sinnvolle Trainer

Danach könnten folgen:

```
Firewall-Regel-Trainer
PNAT-Trainer
TCP/UDP/ICMP-Vergleichstrainer
```