2.6 tcpdump und Capture-Filter – Paketmitschnitte im Terminal

Ziel dieser Seite

`tcpdump` ist ein Kommandozeilenwerkzeug zum Aufzeichnen und Anzeigen von Netzwerkpaketen. Es eignet sich besonders für:

- Linux- und macOS-Systeme ohne grafische Oberfläche;
- Server und Netzwerkgeräte;
- SSH-Sitzungen;
- kurze, gezielte Paketmitschnitte;
- automatisierte oder zeitlich begrenzte Aufzeichnungen;
- die Vorbereitung einer späteren Analyse mit Wireshark oder TShark;
- die Prüfung, ob ein Paket einen bestimmten Aufzeichnungspunkt erreicht.

`tcpdump` verwendet die Capture-Filter-Sprache von `libpcap`. Diese Filter bestimmen bereits während des Mitschnitts, welche Pakete verarbeitet und gespeichert werden.

> **Wichtig:** Ein Capture-Filter lässt sich nachträglich nicht rückgängig machen. Pakete, die nicht zum Filter passen, werden nicht in die Mitschnittdatei aufgenommen.

Kennzeichnungen

Kennzeichnung	Bedeutung
`[RO]`	Nur lesender beziehungsweise auswertender Befehl
`[TEST]`	Aktiver Diagnose- oder Aufzeichnungsvorgang
`[PRIV]`	Erhöhte Berechtigungen können erforderlich sein
`[FILE]`	Der Befehl erstellt oder verarbeitet eine Datei
`[SENS]`	Ausgabe oder Datei kann sensible Informationen enthalten
`[CHANGE]`	Der Befehl verändert eine Einstellung oder Datei
`[DISRUPT]`	Der Vorgang kann Netzwerk oder Systembetrieb beeinflussen

---

1. Auf welchen Betriebssystemen kann tcpdump verwendet werden?

Betriebssystemübersicht anzeigen

Betriebssystem	Verfügbarkeit	Typischer Aufruf
Linux	Meist über die Paketverwaltung installierbar	`sudo tcpdump`
macOS	Standardmäßig vorhanden	`sudo tcpdump`
FreeBSD/OpenBSD/NetBSD	Üblicherweise vorhanden	`doas tcpdump` oder `sudo tcpdump`
Windows	Nicht Bestandteil von Windows	Vorzugsweise TShark oder Wireshark mit Npcap
Netzwerk-Appliance	Häufig vorhanden, aber herstellerabhängig	Herstellerdokumentation beachten
Container	Nur mit passenden Netzwerk- und Capture-Berechtigungen	Meist besser auf dem Container-Host mitschneiden

Version prüfen

```
[RO] tcpdump --version
```

Die Ausgabe nennt normalerweise sowohl die tcpdump- als auch die libpcap-Version.

Linux: Installation prüfen

```
[RO] command -v tcpdump
```

macOS: Installationspfad prüfen

```
[RO] command -v tcpdump
```

Windows

Unter Windows sollte für eine vergleichbare Aufgabe normalerweise TShark verwendet werden:

```
[RO] tshark --version
```

```
[RO] tshark -D
```

WinDump existiert als tcpdump-ähnliches Werkzeug für Windows, ist aber kein Bestandteil der aktuellen Windows-Bordmittel. Für ein modernes Administrationsumfeld sind Wireshark und TShark mit Npcap in der Regel die besser dokumentierte Lösung.

---

2. Welche Berechtigungen benötigt tcpdump?

Capture-Berechtigungen und Sicherheitsprinzip anzeigen

Das Lesen von Netzwerkpaketen erfordert in der Regel besondere Betriebssystemberechtigungen. Ein normaler Benutzer darf deshalb häufig keinen Live-Mitschnitt starten.

Typischer Aufruf unter Linux und macOS

```
[TEST][PRIV][SENS] sudo tcpdump
```

Das Lesen einer bereits vorhandenen Mitschnittdatei benötigt normalerweise keine Capture-Berechtigung:

```
[RO][FILE][SENS] tcpdump -r capture.pcap
```

Sicherheitsprinzip

- tcpdump nur für den erforderlichen Zeitraum mit erhöhten Rechten starten;
- nicht dauerhaft als `root` arbeiten;
- Ausgabedateien in einem geschützten Verzeichnis speichern;
- Dateiberechtigungen nach dem Mitschnitt kontrollieren;
- Paketmitschnitte nicht unkontrolliert weitergeben;
- nur autorisierte Schnittstellen und Systeme untersuchen.

Je nach Linux-Distribution kann tcpdump mit Linux-Capabilities oder einer herstellerspezifischen Berechtigungsregel ausgestattet sein. Solche Änderungen dürfen nicht pauschal aus fremden Anleitungen übernommen werden. Vorher müssen Sicherheitsrichtlinie, Paketquelle und bestehende Berechtigungen geprüft werden.

---

3. Wie werden Schnittstellen ermittelt und ausgewählt?

Schnittstellenbefehle anzeigen

Mit tcpdump verfügbare Capture-Schnittstellen anzeigen

```
[RO] tcpdump -D
```

Alternativ:

```
[RO] tcpdump --list-interfaces
```

Beispielhafte Schnittstellennamen:

System	Typische Namen
Linux	`eth0`, `ens18`, `enp3s0`, `wlan0`, `lo`
macOS	`en0`, `en1`, `lo0`, `utun0`
Container-Host	`docker0`, `br-...`, `veth...`
VPN	`tun0`, `tap0`, `wg0`, `utun...`

Bestimmte Schnittstelle verwenden

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0
```

```
[TEST][PRIV][SENS] sudo tcpdump -i en0
```

Loopbackverkehr aufzeichnen

Linux:

```
[TEST][PRIV][SENS] sudo tcpdump -i lo
```

macOS:

```
[TEST][PRIV][SENS] sudo tcpdump -i lo0
```

Mehrere reguläre Schnittstellen erfassen

```
[TEST][PRIV][SENS] sudo tcpdump -i any
```

Die Pseudoschnittstelle `any` wird auf Linux und aktuellen macOS-Versionen unterstützt. Dabei gelten Einschränkungen:

- der Mitschnitt erfolgt nicht im Promiscuous Mode;
- unterschiedliche Link-Layer-Typen können auftreten;
- die Ausgabe kann mehr Verkehr enthalten als erwartet;
- die konkrete Eingangsschnittstelle sollte bei der Auswertung beachtet werden.

Richtige Schnittstelle kontrollieren

```
[TEST][PRIV][SENS] sudo tcpdump -i en0 -nn -c 20
```

Danach wird ein bekannter, autorisierter Test erzeugt, beispielsweise:

```
ping 192.0.2.1
```

Sind die erwarteten Pakete nicht sichtbar, müssen folgende Punkte geprüft werden:

- falsche Schnittstelle;
- VPN- oder Tunnelschnittstelle übersehen;
- Loopbackverkehr auf physischer Schnittstelle gesucht;
- falscher Netzwerk-Namespace;
- Paket passiert den untersuchten Host nicht;
- Capture-Berechtigung fehlt;
- Capture-Filter ist zu eng.

---

4. Welche Optionen werden bei tcpdump häufig verwendet?

Optionstabelle anzeigen

Option	Bedeutung
`-D`	Verfügbare Capture-Schnittstellen anzeigen
`-i INTERFACE`	Schnittstelle auswählen
`-n`	Adressen und andere numerische Angaben nicht in Namen umwandeln
`-nn`	Insbesondere auch Portnummern numerisch anzeigen
`-c ANZAHL`	Nach einer bestimmten Paketanzahl beenden
`-w DATEI`	Rohpakete in eine Datei schreiben

Option	Bedeutung
`-r DATEI`	Pakete aus einer Datei lesen
`-v`	Ausführlichere Ausgabe
`-vv`	Noch ausführlichere Ausgabe
`-vvv`	Maximale standardmäßige Detailstufe
`-q`	Verkürzte Ausgabe
`-e`	Link-Layer-Header, beispielsweise MAC-Adressen, anzeigen
`-A`	Nutzdaten als ASCII anzeigen
`-x`	Paketdaten hexadezimal anzeigen
`-X`	Paketdaten hexadezimal und als ASCII anzeigen
`-XX`	Wie `-X`, einschließlich Link-Layer-Header
`-tt`	Zeit als Unix-Zeitstempel ausgeben
`-ttt`	Zeitdifferenz zum vorherigen Paket ausgeben
`-tttt`	Datum und Uhrzeit pro Paket ausgeben
`-ttttt`	Zeitdifferenz zum ersten angezeigten Paket ausgeben
`-s LÄNGE`	Maximale Anzahl aufgezeichneter Bytes je Paket festlegen
`-B GRÖSSE`	Betriebssystem-Capture-Puffer in KiB festlegen
`-C GRÖSSE`	Ausgabedatei nach Erreichen einer Größe wechseln
`-G SEKUNDEN`	Ausgabedatei zeitgesteuert wechseln
`-W ANZAHL`	Anzahl rotierter Dateien begrenzen
`-U`	Pakete zeitnah in die Ausgabedatei schreiben
`-K`	Prüfsummenprüfung bei der Anzeige deaktivieren
`-Q in|out|inout`	Aufzeichnungsrichtung festlegen, sofern unterstützt
`-p`	Promiscuous Mode nicht aktivieren
`-F DATEI`	Capture-Filter aus einer Datei lesen

Nicht jede Option wird von jeder Betriebssystem- und libpcap-Version unterstützt. Deshalb sollten Version und lokale Hilfeseite geprüft werden:

```
[RO] tcpdump --version
```

```
[RO] man tcpdump
```

---

5. Wie wird ein kurzer, kontrollierter Mitschnitt durchgeführt?

Grundlegende Mitschnittbefehle anzeigen

20 Pakete anzeigen und danach beenden

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -c 20
```

100 Pakete in eine Datei schreiben

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -c 100 -w capture.pcap
```

Nur Verkehr eines bestimmten Hosts aufzeichnen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -c 500 -w host-test.pcap 'host 192.0.2.10'
```

Nur HTTPS-Verkehr zu einem Server aufzeichnen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -c 1000 -w https-test.pcap 'host 198.51.100.20 and tcp port 443'
```

Mitschnitt manuell beenden

Ein laufender Mitschnitt wird normalerweise mit folgender Tastenkombination beendet:

```
[Strg] + [C]
```

Nach dem Beenden zeigt tcpdump Zähler an:

```
packets captured
packets received by filter
packets dropped by kernel
```

Die genaue Bedeutung von `packets received by filter` ist betriebssystemabhängig. Besonders wichtig ist:

```
packets dropped by kernel
```

Ein Wert größer als null weist darauf hin, dass der Capture-Mechanismus Pakete wegen fehlenden Pufferplatzes verworfen hat. Die Aufzeichnung ist dann möglicherweise unvollständig.

---

6. Wie ist ein tcpdump-Filter grundsätzlich aufgebaut?

BPF-Filterlogik anzeigen

Ein Capture-Filter besteht aus einer oder mehreren Bedingungen. Diese können mit logischen Operatoren verbunden werden.

Operator	Bedeutung
`and`	Beide Bedingungen müssen zutreffen
`or`	Mindestens eine Bedingung muss zutreffen
`not`	Bedingung wird ausgeschlossen
`( )`	Gruppiert mehrere Bedingungen

Einzelne Bedingung

```
host 192.0.2.10
```

Zwei Bedingungen mit `and`

```
host 192.0.2.10 and tcp port 443
```

Alternative Bedingungen mit `or`

```
tcp port 80 or tcp port 443
```

Bedingung ausschließen

```
not port 22
```

Gruppierung mit Klammern

```
host 192.0.2.10 and (tcp port 80 or tcp port 443)
```

Filter mit Leerzeichen, Klammern oder Shell-Sonderzeichen sollten immer in einfache Anführungszeichen gesetzt werden:

```
sudo tcpdump -i eth0 -nn 'host 192.0.2.10 and (tcp port 80 or tcp port 443)'
```

Dadurch wird verhindert, dass die Shell Zeichen wie `(`, `)`, `!`, `&` oder `|` selbst interpretiert.

---

7. Welche Host-, Netz- und Richtungsfilter gibt es?

Host- und Netzfilter anzeigen

Aufgabe	Filter
Host als Quelle oder Ziel	`host 192.0.2.10`
Nur Pakete von einem Host	`src host 192.0.2.10`
Nur Pakete zu einem Host	`dst host 192.0.2.10`
Einer von zwei Hosts	`host 192.0.2.10 or host 198.51.100.20`
Kommunikation zwischen zwei Hosts	`host 192.0.2.10 and host 198.51.100.20`
Host ausschließen	`not host 192.0.2.10`
IPv4-Netz	`net 192.0.2.0/24`
Nur Quelle aus einem Netz	`src net 192.0.2.0/24`
Nur Ziel in einem Netz	`dst net 198.51.100.0/24`
Netz ausschließen	`not net 192.0.2.0/24`
Bestimmte Ethernet-Adresse	`ether host 00:11:22:33:44:55`
Nur Ethernet-Quelle	`ether src 00:11:22:33:44:55`
Nur Ethernet-Ziel	`ether dst 00:11:22:33:44:55`

Kommunikation zwischen genau zwei IPv4-Systemen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'host 192.0.2.10 and host 198.51.100.20'
```

Bei zwei unterschiedlichen Hosts bedeutet diese Schreibweise praktisch, dass beide Adressen im Paket vorkommen müssen: eine als Quelle und die andere als Ziel.

Nur Verkehr vom Client zum Server

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'src host 192.0.2.10 and dst host 198.51.100.20'
```

Nur Rückverkehr

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'src host 198.51.100.20 and dst host 192.0.2.10'
```

---

8. Welche Protokoll- und Portfilter werden häufig verwendet?

Protokoll- und Portfilter anzeigen

Aufgabe	Filter
Nur IPv4	`ip`
Nur IPv6	`ip6`
TCP	`tcp`
UDP	`udp`
ICMPv4	`icmp`
ICMPv6	`icmp6`
ARP	`arp`
TCP- oder UDP-Port 53	`port 53`
Nur TCP-Port 53	`tcp port 53`
Nur UDP-Port 53	`udp port 53`
TCP-Quellport 443	`tcp src port 443`
TCP-Zielport 443	`tcp dst port 443`
Portbereich	`tcp portrange 8000-8100`
Zielportbereich	`tcp dst portrange 8000-8100`
TCP-Port 80 oder 443	`tcp port 80 or tcp port 443`

Aufgabe	Filter
TCP-Port 22 ausschließen	`tcp and not port 22`
Broadcastverkehr	`ether broadcast`
Multicastverkehr	`ether multicast`

Wichtiger Unterschied

```
port 53
```

erfasst TCP und UDP auf Port 53.

```
udp port 53
```

erfasst ausschließlich UDP auf Port 53.

```
tcp port 53
```

erfasst ausschließlich TCP auf Port 53.

Dieser Unterschied ist beispielsweise bei DNS wichtig, da DNS sowohl UDP als auch TCP verwenden kann.

---

9. Welche Filter eignen sich für typische Netzwerkprotokolle?

Protokolltabelle anzeigen

Protokoll oder Aufgabe	Capture-Filter
ARP	`arp`
ICMPv4	`icmp`
ICMPv6	`icmp6`
DNS über TCP und UDP	`port 53`
DHCPv4	`udp port 67 or udp port 68`
DHCPv6	`udp port 546 or udp port 547`
HTTP	`tcp port 80`

Protokoll oder Aufgabe	Capture-Filter
HTTPS	`tcp port 443`
HTTP und HTTPS	`tcp port 80 or tcp port 443`
SSH	`tcp port 22`
SMTP	`tcp port 25`
NTP	`udp port 123`
SNMP	`udp port 161 or udp port 162`
LDAP	`tcp port 389 or udp port 389`
LDAPS	`tcp port 636`
Kerberos	`port 88`
SMB	`tcp port 445`
RDP	`tcp port 3389 or udp port 3389`
SIP	`port 5060 or port 5061`
TFTP	`udp port 69`
WireGuard	Üblicherweise der konfigurierte UDP-Port, beispielsweise `udp port 51820`
IPsec IKE	`udp port 500 or udp port 4500`
VXLAN-Standardport	`udp port 4789`

DNS-Aufzeichnung

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w dns.pcap 'port 53'
```

DHCPv4-Aufzeichnung

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -e -w dhcp.pcap 'udp port 67 or udp port 68'
```

SMB-Kommunikation mit einem Server

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w smb-test.pcap 'host 198.51.100.20 and tcp port 445'
```

> Portfilter zeigen Verkehr auf den angegebenen Ports. Sie beweisen nicht automatisch, dass der Verkehr tatsächlich das erwartete

> Anwendungsprotokoll enthält.

---

10. Wie werden TCP-Verbindungsaufbau und Verbindungsabbruch gefiltert?

TCP-Flag-Filter anzeigen

Die TCP-Steuerflags befinden sich im TCP-Header. Mit `tcp[tcpflags]` lassen sie sich gezielt prüfen.

Aufgabe	Capture-Filter
Alle Pakete mit gesetztem SYN	`tcp[tcpflags] & tcp-syn != 0`
Initiales SYN ohne ACK	`tcp[tcpflags] & (tcp-syn`
SYN und ACK gesetzt	`tcp[tcpflags] & (tcp-syn`
Pakete mit RST	`tcp[tcpflags] & tcp-rst != 0`
Pakete mit FIN	`tcp[tcpflags] & tcp-fin != 0`
Pakete mit PSH	`tcp[tcpflags] & tcp-push != 0`

Neue TCP-Verbindungsversuche anzeigen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp[tcpflags] & (tcp-syn|tcp-ack) == tcp-syn'
```

SYN/ACK-Antworten anzeigen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp[tcpflags] & (tcp-syn|tcp-ack) == (tcp-syn|tcp-ack)'
```

TCP-Resets anzeigen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp[tcpflags] & tcp-rst != 0'
```

Nur TCP-Resets eines bestimmten Servers

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'host 198.51.100.20 and tcp[tcpflags] & tcp-rst != 0'
```

Interpretation

Beobachtung	Mögliche Bedeutung
Wiederholte SYN-Pakete ohne SYN/ACK	Keine Antwort sichtbar, Paketverlust, Firewall, Routing- oder Serverproblem
SYN wird mit RST beantwortet	System erreichbar, aber Port geschlossen oder Verbindung abgelehnt
SYN/ACK erreicht Client nicht	Rückweg, Firewall, asymmetrisches Routing oder Aufzeichnungspunkt prüfen
RST während bestehender Verbindung	Endpunkt oder Zwischensystem bricht die TCP-Verbindung ab
FIN-Sequenz	Regulärer geordneter Verbindungsabbau möglich

tcpdump kennzeichnet Wiederholungsübertragungen nicht so komfortabel wie Wireshark. Für eine detaillierte TCP-Analyse sollte die Datei anschließend mit Wireshark oder TShark untersucht werden.

---

11. Wie werden Paketgrößen gefiltert?

Längenfilter anzeigen

Aufgabe	Filter
Pakete kleiner oder gleich 100 Byte	`less 100`
Pakete größer oder gleich 1500 Byte	`greater 1500`
TCP-Pakete größer oder gleich 1500 Byte	`tcp and greater 1500`
UDP-Pakete kleiner oder gleich 100 Byte	`udp and less 100`

Große IPv4-Pakete anzeigen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'ip and greater 1500'
```

Bei `less` und `greater` ist zu beachten:

- `less N` bedeutet kleiner oder gleich `N`;
- `greater N` bedeutet größer oder gleich `N`;
- die Paketlänge allein beweist noch kein MTU-Problem;
- VLAN-, Tunnel- und Link-Layer-Header beeinflussen die beobachtete Größe;
- Fragmentierung und Path-MTU-Discovery müssen getrennt untersucht werden.

---

12. Wie werden VLAN-Pakete gefiltert?

VLAN-Filter und Fallstricke anzeigen

Pakete mit 802.1Q-VLAN-Tag

```
vlan
```

Bestimmte VLAN-ID

```
vlan 100
```

IPv4-Verkehr im VLAN 100

```
vlan 100 and ip
```

HTTPS-Verkehr im VLAN 100

```
vlan 100 and tcp port 443
```

Aufzeichnung

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -e -w vlan100.pcap 'vlan 100'
```

Die Option `-e` zeigt bei der Textausgabe den Link-Layer-Header an und kann dadurch VLAN- und MAC-Informationen sichtbar machen.

Wichtige Einschränkungen

- Netzwerkkarten können VLAN-Tags per Hardware-Offloading entfernen oder separat an das Betriebssystem übergeben.
- Ein auf dem Endgerät fehlender VLAN-Tag beweist deshalb nicht automatisch, dass auf dem Kabel kein VLAN-Tag vorhanden war.
- Access-Ports liefern dem Endgerät normalerweise ungetaggte Frames.
- Trunk-Ports können mehrere getaggte VLANs transportieren.
- Bei mehrfach getaggten Frames kann ein weiterer `vlan`-Ausdruck erforderlich sein.

Beispiel für doppelte VLAN-Kapselung:

```
vlan 100 and vlan 200
```

Ob diese Pakete am gewählten Aufzeichnungspunkt sichtbar sind, hängt von Netzwerkkarte, Treiber, Offloading und Switch-Konfiguration ab.

---

13. Wie wird die Textausgabe sinnvoll formatiert?

Ausgabeoptionen und Beispiele anzeigen

Numerische Ausgabe mit Datum und Uhrzeit

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -tttt
```

MAC-Adressen anzeigen

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -e
```

Ausführliche Ausgabe

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -vv
```

Hexadezimal- und ASCII-Ausgabe

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -X -c 10
```

Link-Layer-Header, Hexadezimal- und ASCII-Daten

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -XX -c 10
```

Kurzausgabe

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -q
```

Bedeutung der Zeitoptionen

Option	Ausgabe
Keine zusätzliche Option	Uhrzeit seit Mitternacht
`-tt`	Sekunden seit Unix-Epoche
`-ttt`	Differenz zum vorherigen Paket
`-tttt`	Datum und Uhrzeit
`-ttttt`	Differenz zum ersten Paket

> `-A`, `-X` und `-XX` können sensible Nutzdaten sichtbar machen. Sie sollten nur gezielt und bei entsprechender Berechtigung verwendet werden.

---

14. Wie werden Mitschnittdateien gespeichert und gelesen?

Dateibefehle anzeigen

Pakete als Rohdaten speichern

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w capture.pcap
```

Während `-w` verwendet wird, schreibt tcpdump die Rohpakete in die Datei. Die normale Paketbeschreibung erscheint dabei nicht automatisch im Terminal.

Vorhandene Datei lesen

```
[RO][FILE][SENS] tcpdump -nn -r capture.pcap
```

Mit Datum und ausführlicher Ausgabe lesen

```
[RO][FILE][SENS] tcpdump -nn -tttt -vv -r capture.pcap
```

Nur DNS aus einer vorhandenen Datei lesen

```
[RO][FILE][SENS] tcpdump -nn -r capture.pcap 'port 53'
```

Nur Verkehr eines Hosts lesen

```
[RO][FILE][SENS] tcpdump -nn -r capture.pcap 'host 192.0.2.10'
```

Paketanzahl einer Datei bestimmen, sofern von der installierten Version unterstützt

```
[RO][FILE][SENS] tcpdump --count -r capture.pcap
```

Alternativ kann das Wireshark-Werkzeug Capinfos verwendet werden:

```
[RO][FILE][SENS] capinfos capture.pcap
```

Datei während des Mitschnitts paketweise aktualisieren

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -U -w capture.pcap
```

`-U` sorgt dafür, dass jedes empfangene Paket zeitnah in die Ausgabedatei geschrieben wird. Das kann hilfreich sein, wenn eine Aufzeichnung während eines Absturzes oder Fehlers möglichst aktuell bleiben soll.

---

15. Wie wird die Größe eines Mitschnitts begrenzt?

Dateigröße und Ringpuffer anzeigen

Nach ungefähr 100 MB eine neue Datei beginnen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -C 100 -w capture.pcap
```

Ohne Einheit verwendet `-C` Millionen Byte:

```
100 = 100.000.000 Byte
```

Moderne tcpdump-Versionen unterstützen zusätzlich Einheiten:

Angabe	Einheit
`100K`	100 × 1.024 Byte
`100M`	100 × 1.048.576 Byte

Angabe	Einheit
`1G`	1 × 1.073.741.824 Byte

Zehn Dateien mit jeweils ungefähr 100 MiB als Ringpuffer

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -C 100M -W 10 -w capture.pcap
```

Bei Verwendung von `-C` zusammen mit `-W`:

- wird nach Erreichen der Dateigröße zur nächsten Datei gewechselt;
- wird die Anzahl der Dateien begrenzt;
- beginnt tcpdump nach der letzten Datei wieder mit der ersten;
- werden ältere Dateien überschrieben;
- entsteht ein echter rotierender Ringpuffer.

Ungefähre maximale Gesamtgröße

```
10 Dateien × 100 MiB = ungefähr 1.000 MiB
```

Die Größenprüfung erfolgt vor dem Schreiben eines neuen Pakets. Eine Datei kann deshalb geringfügig größer als der angegebene Wert werden.

---

16. Wie werden Mitschnittdateien zeitgesteuert gewechselt?

Zeitrotation korrekt verwenden

Alle fünf Minuten eine neue Datei

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -G 300 -w 'capture-%Y%m%d-%H%M%S.pcap'
```

Beispielhafte Dateinamen:

```
capture-20260731-140000.pcap
capture-20260731-140500.pcap
capture-20260731-141000.pcap
```

Nach zwölf Fünf-Minuten-Dateien beenden

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -G 300 -W 12 -w 'capture-%Y%m%d-%H%M%S.pcap'
```

Das entspricht ungefähr:

```
12 Dateien × 5 Minuten = 60 Minuten
```

Wichtiger Unterschied

Kombination	Verhalten
`-C` und `-W`	Ringpuffer; ältere Dateien werden überschrieben
`-G` und `-W`	Nach der festgelegten Anzahl Dateien wird tcpdump beendet
`-C`, `-G` und `-W` gemeinsam	`-W` begrenzt nach aktueller Dokumentation nicht zuverlässig wie bei den einzelnen Varianten und beeinflusst hauptsächlich die Dateinamen

Bei `-G` sollte `-w` einen eindeutigen Zeitplatzhalter enthalten. Andernfalls kann jede neue Datei denselben Namen erhalten und die vorherige Datei überschreiben.

Sicheres Beispiel

```
-w 'capture-%Y%m%d-%H%M%S.pcap'
```

Problematisches Beispiel

```
-w capture.pcap
```

---

17. Wie wird die Aufzeichnungsdauer auf Linux und macOS begrenzt?

Zeitlich begrenzte Aufzeichnung anzeigen

tcpdump besitzt die Option `-G` für den Dateiw echsel, aber keine allgemeine, plattformübergreifende Option nach dem Muster „nach genau 60 Sekunden stoppen“.

Linux mit GNU `timeout`

```
[TEST][PRIV][FILE][SENS] sudo timeout 60 tcpdump -i eth0 -nn -w capture.pcap 'host 192.0.2.10'
```

`timeout` gehört nicht zu tcpdump, sondern zu den GNU Coreutils. Der Befehl steht nicht auf jedem Unix-System standardmäßig zur Verfügung.

macOS

Auf macOS ist GNU `timeout` standardmäßig nicht vorhanden. Stattdessen sind folgende Methoden geeignet:

- mit `-c` eine maximale Paketanzahl festlegen;
- mit `-G` und `-W 1` genau eine zeitlich begrenzte Datei erstellen;
- den Mitschnitt kontrolliert mit `[Strg] + [C]` beenden.

macOS: nach einem Zeitintervall beenden

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i en0 -nn -G 60 -W 1 -w 'capture-%Y%m%d-%H%M%S.pcap' 'host 192.0.2.10'
```

Dieser Befehl erstellt eine Aufzeichnungsdatei und beendet tcpdump nach dem ersten Rotationsintervall.

> Vor dem praktischen Einsatz sollte das Verhalten der lokal installierten tcpdump-Version mit einem kurzen Test geprüft werden.

---

18. Wie wird die Snapshot-Länge verwendet?

Snapshot-Länge und Auswirkungen anzeigen

Die Snapshot-Länge bestimmt, wie viele Byte eines Pakets gespeichert werden.

```
-s LÄNGE
```

Beispiel:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -s 128 -w headers.pcap
```

Damit werden höchstens 128 Byte je Paket aufgezeichnet.

Vorteile einer verkürzten Snapshot-Länge

- kleinere Mitschnittdateien;
- weniger Nutzdaten werden gespeichert;
- geringerer Speicherbedarf;
- bei sehr hohem Datenaufkommen möglicherweise weniger Verarbeitungsaufwand.

Nachteile

- Protokollinformationen können abgeschnitten werden;
- Anwendungsdaten fehlen;
- Pakete lassen sich möglicherweise nicht vollständig analysieren;
- VLAN-, Tunnel- und umfangreiche Protokollheader können mehr Platz benötigen;
- Wireshark zeigt möglicherweise abgeschnittene Pakete an.

Abgeschnittene Pakete können in der tcpdump-Ausgabe mit einer Kennzeichnung wie dieser erscheinen:

```
[|proto]
```

Die aktuelle tcpdump-Standard-Snapshot-Länge ist sehr groß und erfasst normalerweise das vollständige Paket. Eine kleinere Snapshot-Länge sollte nur verwendet werden, wenn klar ist, welche Header und Daten für die Diagnose benötigt werden.

---

19. Wie werden Paketverluste während des Mitschnitts erkannt?

Capture-Drops und Gegenmaßnahmen anzeigen

Beim Beenden zeigt tcpdump normalerweise unter anderem:

```
packets captured
packets received by filter
packets dropped by kernel
```

Bewertung

Zähler	Bedeutung
`packets captured`	Von tcpdump empfangene und verarbeitete Pakete

Zähler	Bedeutung
`packets received by filter`	Betriebssystemabhängiger Zähler des Capture-Mechanismus
`packets dropped by kernel`	Wegen fehlenden Capture-Pufferplatzes verworfene Pakete

Sind Kernel-Drops vorhanden, können folgende Maßnahmen helfen:

1. Capture-Filter enger setzen.
2. Mitschnitt auf eine Datei schreiben statt alles im Terminal auszugeben.
3. Ausgabeoptionen wie `-A`, `-X`, `-XX` oder `-vvv` während des Live-Mitschnitts vermeiden.
4. Capture-Puffer vorsichtig vergrößern.
5. Snapshot-Länge nur bei fachlicher Begründung verkleinern.
6. Auf einem leistungsfähigeren oder geeigneteren System mitschneiden.
7. Festplattenleistung und freien Speicherplatz prüfen.
8. Netzwerk-TAP oder spezielle Capture-Hardware verwenden.

Capture-Puffer beispielsweise auf 4.096 KiB setzen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -B 4096 -w capture.pcap
```

Ein größerer Puffer garantiert keinen verlustfreien Mitschnitt. Der Wert muss zum Betriebssystem, Datenaufkommen und verfügbaren Speicher passen.

---

20. Warum zeigt tcpdump manchmal fehlerhafte Prüfsummen an?

Checksum-Offloading erklären

Netzwerkkarten können Prüfsummenberechnungen für IP, TCP oder UDP übernehmen. Bei ausgehenden Paketen kann tcpdump das Paket erfassen, bevor die Netzwerkkarte die endgültige Prüfsumme eingesetzt hat.

Dadurch kann tcpdump scheinbar fehlerhafte Prüfsummen anzeigen, obwohl das Paket korrekt übertragen wurde.

Prüfsummenprüfung bei der Anzeige deaktivieren

```
[RO][FILE][SENS] tcpdump -K -nn -r capture.pcap
```

Live:

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -K -nn
```

`-K` verhindert lediglich die Prüfsummenprüfung durch tcpdump. Es repariert keine Pakete und verändert nicht die Netzwerkkonfiguration.

Sinnvolle Prüfung

- Mitschnitt auf der empfangenden Gegenstelle vergleichen;
- Netzwerkpfad oder TAP untersuchen;
- Offloading-Einstellungen dokumentieren;
- einen angezeigten Prüfsummenfehler nicht ohne weitere Prüfung als Netzwerkdefekt bewerten.

Das Deaktivieren von Hardware-Offloading wäre eine Systemänderung und kann Leistung oder Verhalten beeinflussen. Es gehört deshalb nicht zu den ersten Diagnosemaßnahmen.

---

21. Wie wird eingehender oder ausgehender Verkehr gefiltert?

Richtungsfilter anzeigen

Auf unterstützten Plattformen kann `-Q` verwendet werden:

Option	Bedeutung
`-Q in`	Nur eingehender Verkehr
`-Q out`	Nur ausgehender Verkehr
`-Q inout`	Beide Richtungen

Nur eingehende Pakete

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -Q in -nn
```

Nur ausgehende DNS-Pakete

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -Q out -nn 'port 53'
```

`-Q` wird nicht auf jeder Plattform und nicht von jeder Capture-Schnittstelle unterstützt.

Eine portablere Alternative besteht darin, Quell- und Zieladresse ausdrücklich zu filtern:

```
src host 192.0.2.10
```

oder:

```
dst host 192.0.2.10
```

Dabei wird die Richtung anhand der Paketadressen und nicht anhand der Betriebssystem-Schnittstellenrichtung bestimmt.

---

22. Wie wird ein Filter vor dem eigentlichen Mitschnitt geprüft?

Filtertest und Fehlermeldungen anzeigen

Ein falsch formulierter Filter kann dazu führen, dass tcpdump nicht startet oder unerwartete Pakete erfasst.

Kurzer Praxistest

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -c 10 'host 192.0.2.10 and tcp port 443'
```

Kompilierten BPF-Code anzeigen

```
[RO][PRIV] sudo tcpdump -i eth0 -d 'host 192.0.2.10 and tcp port 443'
```

`-d` kompiliert den Filter, gibt die resultierenden BPF-Anweisungen lesbar aus und beendet tcpdump, ohne einen normalen Mitschnitt zu starten.

Filter aus einer Datei lesen

Beispielinhalt einer autorisiert erstellten Filterdatei:

```
host 192.0.2.10 and
(tcp port 80 or tcp port 443)
```

Filterdatei verwenden:

```
[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -F capture-filter.txt
```

Wird `-F` verwendet, wird ein zusätzlicher Filterausdruck auf der Kommandozeile ignoriert. Deshalb sollte nicht gleichzeitig ein weiterer Filter hinter dem Befehl angegeben werden.

---

23. Welche typischen Fehler treten bei tcpdump auf?

Fehlertabelle anzeigen

Fehler oder Beobachtung	Wahrscheinliche Erklärung	Prüfung
`tcpdump: command not found`	tcpdump nicht installiert oder nicht im Suchpfad	`command -v tcpdump`
`You don't have permission to capture`	Capture-Berechtigung fehlt	Berechtigungen und Sicherheitsrichtlinie prüfen
Keine Pakete sichtbar	Falsche Schnittstelle oder zu enger Filter	`tcpdump -D`, Kontrollmitschnitt ohne Filter
Namen statt IP-Adressen	Namensauflösung aktiv	`-nn` verwenden
Mitschnittdatei wächst unbegrenzt	Keine Größen- oder Zeitbegrenzung	`-C`, `-W` oder `-G` verwenden
Nur eine Datei trotz `-G`	Dateiname enthält keinen eindeutigen Zeitplatzhalter	Zeitformat in `-w` ergänzen
Alte Dateien werden überschrieben	Ringpuffer aktiv oder Dateiname nicht eindeutig	Optionen und Zielverzeichnis prüfen
Hohe Anzahl `dropped by kernel`	Capture-Puffer oder Systemleistung reicht nicht	Filter, `-B`, Ausgabe und Speicher prüfen
`bad udp cksum` oder ähnliche Meldung	Möglicherweise Checksum Offloading	Gegenstelle vergleichen oder Anzeige mit `-K` prüfen
Pakete erscheinen abgeschnitten	Snapshot-Länge zu klein	`-s`-Einstellung prüfen
VLAN-Tag fehlt	Access-Port oder Hardware-Offloading	Switch-Port, Treiber und Gegenmitschnitt prüfen
Fremder Unicastverkehr fehlt	Switch sendet ihn nicht an diesen Port	Autorisierten SPAN-Port oder TAP verwenden
Filter funktioniert in Wireshark, aber nicht in tcpdump	Display-Filter mit Capture-Filter verwechselt	libpcap-Syntax verwenden
SSH-Mitschnitt enthält überwiegend eigene Sitzung	Administrationsverbindung wird mitgeschnitten	Eigene SSH-Verbindung gezielt ausschließen

Eigene SSH-Sitzung ausschließen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w capture.pcap 'not tcp port 22'
```

Dieser Filter schließt allerdings sämtlichen TCP-Verkehr auf Port 22 aus. Falls SSH selbst untersucht werden soll, darf dieser Filter nicht verwendet werden.

---

24. Wie wird tcpdump über SSH sicher eingesetzt?

Remote-Diagnose anzeigen

Beim Start von tcpdump über SSH wird auch die eigene SSH-Verbindung sichtbar, sofern sie den untersuchten Netzwerkpfad verwendet.

Eigene SSH-Verbindung ausschließen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w capture.pcap 'not tcp port 22'
```

Besser ist ein genauer Ausschluss, wenn die Managementadresse bekannt ist:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -w capture.pcap 'not (host 192.0.2.50 and tcp port 22)'
```

Dabei ist `192.0.2.50` beispielhaft die autorisierte Managementstation.

Risiken einer Remote-Sitzung

- die eigene SSH-Verbindung kann den Mitschnitt vergrößern;
- die Mitschnittdatei kann auf einem produktiven Server Speicherplatz belegen;
- ein Verbindungsabbruch kann einen unbegrenzt gestarteten Mitschnitt weiterlaufen lassen;
- eine Shell-Pipe kann sensible Paketdaten über die Administrationsverbindung übertragen.

Deshalb sollten mindestens eine dieser Begrenzungen verwendet werden:

- Paketanzahl mit `-c`;
- Dateigröße mit `-C`;
- Ringpuffer mit `-C` und `-W`;
- zeitliche Begrenzung mit `-G` und `-W`;
- enger Capture-Filter.

Nach der Diagnose muss geprüft werden, ob tcpdump noch läuft:

```
[RO] pgrep -a tcpdump
```

Dieser Befehl ist auf Linux üblich. Die genaue Verfügbarkeit ist betriebssystemabhängig.

---

25. Wie werden Container- und VM-Verbindungen untersucht?

Virtuelle Netzwerke und Namespaces anzeigen

Bei Containern und virtuellen Maschinen kann der Verkehr an mehreren Stellen sichtbar sein:

```
Anwendung
   ↓
Container- oder Gast-Schnittstelle
   ↓
virtuelle Bridge oder virtueller Switch
   ↓
Host-Schnittstelle
   ↓
physisches Netzwerk
```

Typische Linux-Schnittstellen

Schnittstelle	Bedeutung
`docker0`	Standard-Docker-Bridge
`br-...`	Benutzerdefinierte Docker-Bridge
`veth...`	Virtuelles Ethernet-Paar eines Containers
`virbr0`	Häufige libvirt-Bridge
`tap...`	Virtuelle TAP-Schnittstelle
`eth0` oder `ens...`	Physische oder virtuelle Host-Schnittstelle

Schnittstellen anzeigen

```
[RO] ip link show
```

```
[RO] tcpdump -D
```

Docker-Bridge untersuchen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i docker0 -nn -w docker-bridge.pcap
```

Host-Schnittstelle und Container-IP filtern

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i any -nn -w container-test.pcap 'host 192.0.2.10'
```

Ein Mitschnitt innerhalb eines Containers benötigt zusätzliche Berechtigungen wie Netzwerk-Capabilities und erweitert die Angriffsfläche. Wenn möglich, sollte deshalb auf dem autorisierten Container-Host oder an der Bridge aufgezeichnet werden.

Bei Kubernetes, Overlay-Netzen, VXLAN und Service-Proxys kann die sichtbare Adresse je nach Aufzeichnungspunkt durch NAT oder Kapselung verändert sein.

---

26. Wie sieht ein systematischer Diagnoseablauf mit tcpdump aus?

Empfohlene Schrittfolge anzeigen

Vorbereitung

1. Störung, Sollzustand und Fehlerzeitpunkt dokumentieren.
2. Client, Server, IP-Adressen, Ports und Protokolle ermitteln.
3. Den erwarteten Netzwerkpfad bestimmen.
4. Berechtigung für den Mitschnitt prüfen.
5. Systemzeit und Zeitzone kontrollieren.
6. Freien Speicherplatz prüfen.
7. Schutzbedarf der Mitschnittdatei festlegen.

Schnittstelle prüfen

8. Verfügbare Schnittstellen mit `tcpdump -D` anzeigen.
9. Einen kurzen Kontrollmitschnitt ohne engen Filter durchführen.
10. Mit einem bekannten Test prüfen, ob die richtige Schnittstelle ausgewählt wurde.

Mitschnitt planen

11. Einen möglichst gezielten Capture-Filter formulieren.

12. Filter mit einer kleinen Paketanzahl testen.
13. Paket-, Größen- oder Zeitbegrenzung festlegen.
14. Dateinamen, Zielverzeichnis und Berechtigungen prüfen.

Mitschnitt durchführen

15. tcpdump starten.
16. Fehler genau einmal reproduzieren.
17. Start- und Endzeit notieren.
18. Mitschnitt kontrolliert beenden.
19. Capture-Zähler und Kernel-Drops prüfen.
20. Kontrollieren, ob tcpdump noch läuft.

Analyse

21. Datei zunächst mit `tcpdump -nn -r` prüfen.
22. Dateieigenschaften mit `capinfos` kontrollieren.
23. Datei anschließend in Wireshark oder TShark öffnen.
24. Kommunikationsrichtung, TCP-Handshake, DNS, TLS und Anwendung untersuchen.
25. Zeitstempel mit System-, Firewall- und Anwendungslogs vergleichen.

Validierung

26. Beobachtung und vermutete Ursache getrennt dokumentieren.
27. Hypothese durch einen gezielten Vergleichstest prüfen.
28. Falls erforderlich, an einem zweiten Punkt mitschneiden.
29. Nach einer Änderung denselben Test erneut durchführen.
30. Nicht mehr benötigte Mitschnittdateien gemäß Richtlinie löschen.

---

27. Kurzreferenz – tcpdump-Befehle

Befehlstabelle anzeigen

Aufgabe	Befehl
Version anzeigen	`[RO] tcpdump --version`
Schnittstellen anzeigen	`[RO] tcpdump -D`
20 Pakete anzeigen	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -c 20`
100 Pakete speichern	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -c 100 -w capture.pcap`
Alle regulären Schnittstellen	`[TEST][PRIV][SENS] sudo tcpdump -i any -nn`
Datei lesen	`[RO][FILE][SENS] tcpdump -nn -r capture.pcap`

Aufgabe	Befehl
Datei ausführlich lesen	`[RO][FILE][SENS] tcpdump -nn -tttt -vv -r capture.pcap`
MAC-Adressen anzeigen	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -e`
Hex und ASCII anzeigen	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn -X -c 10`
Host filtern	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'host 192.0.2.10'`
HTTPS filtern	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp port 443'`
DNS filtern	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'port 53'`
Initiale TCP-SYN-Pakete	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp[tcpflags] & (tcp-syn|tcp-ack) == tcp-syn'`
TCP-RST-Pakete	`[TEST][PRIV][SENS] sudo tcpdump -i eth0 -nn 'tcp[tcpflags] & tcp-rst != 0'`
Ringpuffer nach Größe	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -C 100M -W 10 -w capture.pcap`
Dateien alle fünf Minuten	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -G 300 -w 'capture-%Y%m%d-%H%M%S.pcap'`
Nach zwölf Zeitdateien stoppen	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -G 300 -W 12 -w 'capture-%Y%m%d-%H%M%S.pcap'`
Capture-Puffer setzen	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -B 4096 -w capture.pcap`
Datei sofort aktualisieren	`[TEST][PRIV][FILE][SENS] sudo tcpdump -i eth0 -nn -U -w capture.pcap`

---

28. Kurzreferenz – Capture-Filter

Filtertabelle anzeigen

Aufgabe	Filter
IPv4	`ip`
IPv6	`ip6`
TCP	`tcp`
UDP	`udp`
ARP	`arp`
ICMPv4	`icmp`
ICMPv6	`icmp6`
Host	`host 192.0.2.10`
Quellhost	`src host 192.0.2.10`

Aufgabe	Filter
Zielhost	`dst host 192.0.2.10`
IPv4-Netz	`net 192.0.2.0/24`
Ethernet-Adresse	`ether host 00:11:22:33:44:55`
TCP- oder UDP-Port	`port 53`
TCP-Port	`tcp port 443`
UDP-Port	`udp port 123`
Zielport	`tcp dst port 443`
Portbereich	`tcp portrange 8000-8100`
DHCPv4	`udp port 67 or udp port 68`
DHCPv6	`udp port 546 or udp port 547`
HTTP oder HTTPS	`tcp port 80 or tcp port 443`
Initiales TCP-SYN	`tcp[tcpflags] & (tcp-syn|tcp-ack) == tcp-syn`
TCP-RST	`tcp[tcpflags] & tcp-rst != 0`
VLAN 100	`vlan 100`
Broadcast	`ether broadcast`
Multicast	`ether multicast`
Große Pakete	`greater 1500`
SSH ausschließen	`not tcp port 22`
Host und HTTPS	`host 192.0.2.10 and tcp port 443`
Host und zwei Ports	`host 192.0.2.10 and (tcp port 80 or tcp port 443)`

Merksätze

- `tcpdump` verwendet Capture-Filter und keine Wireshark-Display-Filter.
- Capture-Filter bestimmen, welche Pakete überhaupt verarbeitet und gespeichert werden.
- Filterausdrücke sollten in einfache Anführungszeichen gesetzt werden.
- `-nn` verhindert störende Namens- und Dienstauflösungen.
- `-w` schreibt Rohpakete und keinen normalen Textbericht.
- `-r` liest eine vorhandene Mitschnittdatei.
- `-C` zusammen mit `-W` erzeugt einen größenbasierten Ringpuffer.
- `-G` zusammen mit `-W` beendet tcpdump nach der festgelegten Anzahl Zeitdateien und ist kein Ringpuffer.
- `packets dropped by kernel` weist auf einen möglicherweise unvollständigen Mitschnitt hin.
- Scheinbar fehlerhafte Prüfsummen können durch Hardware-Offloading entstehen.

- Ein fehlendes Paket ist nur aussagekräftig, wenn Schnittstelle, Aufzeichnungspunkt und Filter nachweislich richtig waren.
- Paketmitschnitte sind sensible Diagnosedaten und müssen entsprechend geschützt werden.

Quellen

- Offizielles tcpdump-Projekt
- Offizieller tcpdump-Quellcode und Dokumentation
- Offizielle tcpdump-Manpage im Quellcode
- Offizielles libpcap-Projekt
- Offizielle pcap-filter-Manpage im libpcap-Quellcode
- Wireshark User’s Guide – Capture Filters
- Offizielle TShark-Dokumentation
- Offizielle Capinfos-Dokumentation
