# 2.14 Sichere Paketmitschnitte, Datenschutz und Beweissicherung

Ein Paketmitschnitt zeichnet Netzwerkpakete an einer Netzwerkschnittstelle oder einem definierten Messpunkt auf. Je nach Protokoll und Verschlüsselung kann eine Aufzeichnung nicht nur technische Metadaten, sondern auch Kommunikationsinhalte enthalten.

Ein Paketmitschnitt kann unter anderem sichtbar machen:

- Quell- und Ziel-IP-Adressen,
- MAC-Adressen,
- Ports und Protokolle,
- DNS-Anfragen und aufgerufene Hostnamen,
- Verbindungsaufbau und Verbindungsabbau,
- Paketgrößen und Zeitstempel,
- TCP-Wiederholungen und Paketverluste,
- unverschlüsselte Benutzernamen oder Inhalte,
- HTTP-Header und Cookies,
- Authentifizierungsinformationen,
- interne System- und Dienstnamen,
- bei vorhandenen Schlüsseln auch entschlüsselte Anwendungsdaten.

> **Grundregel:** So wenig wie möglich, so gezielt wie nötig und nur so lange wie erforderlich mitschneiden.

---

## 1. Kennzeichnungen und Sicherheitsregeln

| Kennzeichnung | Bedeutung |
|---|---|
| [RO] | Liest vorhandene Informationen aus |
| [TEST] | Führt eine aktive Messung oder Aufzeichnung durch |
| [PRIV] | Benötigt möglicherweise Administrator- oder Root-Rechte |
| [FILE] | Erzeugt oder verändert eine Datei |
| [SENS] | Datei oder Ausgabe kann vertrauliche Daten enthalten |
| [CHANGE] | Verändert Konfiguration oder Systemzustand |
| [DISRUPT] | Kann Verbindungen oder Dienste beeinträchtigen |

Ein Live-Paketmitschnitt ist keine rein passive Read-only-Aktion. Er liest laufende Kommunikation mit und schreibt sie in eine Datei:

```
[TEST][PRIV][FILE][SENS]
```

Vor jeder Aufzeichnung müssen mindestens folgende Punkte geklärt werden:

1. Gibt es einen konkreten technischen Anlass?
2. Liegt eine ausreichende betriebliche und rechtliche Freigabe vor?
3. Welche Systeme und Kommunikationsverbindungen dürfen erfasst werden?
4. Wer ist für die Aufzeichnung verantwortlich?
5. Welcher Zeitraum ist erforderlich?
6. Welche Daten sollen ausdrücklich nicht erfasst werden?
7. Wo wird die Datei gespeichert?
8. Wer darf die Datei auswerten?
9. Darf sie an Hersteller oder Dienstleister weitergegeben werden?
10. Wann wird sie gelöscht?

## 2. Warum sind Paketmitschnitte besonders sensibel?

Eine PCAP- oder PCAPNG-Datei kann mehr Informationen enthalten, als für die technische Untersuchung zunächst sichtbar sind.

Mögliche Inhalte:

| Datenart | Beispiel |
|---|---|
| Netzwerkkennungen | IP- und MAC-Adressen |
| Benutzerbezug | Benutzername, Gerät oder interne Zuordnung |
| Kommunikationspartner | interne und externe Zielsysteme |
| DNS-Daten | angefragte Domains und Hostnamen |
| Zeitdaten | Zeitpunkt und Dauer der Kommunikation |
| Inhaltsdaten | unverschlüsselte Nachrichten oder Dateien |
| Authentifizierungsdaten | Cookies, Tokens oder Anmeldeinformationen |
| Organisationsdaten | interne Server-, Standort- und Kundennamen |
| Sicherheitsdaten | Ports, Dienste, Zertifikate und Netzwerkstruktur |
| Entschlüsselungsgeheimnisse | TLS-Key-Log oder eingebettete Schlüsselblöcke |

Auch verschlüsselter Datenverkehr enthält auswertbare Metadaten:

- beteiligte IP-Adressen,
- Verbindungszeiten,
- Datenmengen,
- Paketgrößen,
- DNS-Anfragen, sofern nicht verschlüsselt,
- TLS-Versionen und Zertifikatsinformationen,
- teilweise Servernamen,
- zeitliche Kommunikationsmuster.

Verschlüsselung macht einen Paketmitschnitt daher nicht automatisch datenschutzrechtlich oder sicherheitstechnisch unbedenklich.

## 3. Welche betrieblichen und rechtlichen Prüfungen sind erforderlich?

Die zulässige Verarbeitung hängt vom konkreten Zweck, der Organisation, den betroffenen Personen und den geltenden Vorgaben ab. Diese Seite ersetzt keine rechtliche Prüfung.

Möglicherweise einzubeziehende Stellen:

- System- oder Serviceverantwortliche,
- Informationssicherheitsbeauftragte,
- Datenschutzbeauftragte,
- Netzwerkverantwortliche,
- betroffene Fachabteilung,
- Incident-Response-Team,
- Betriebs- oder Personalrat,
- externe Auftragsverarbeiter,
- Rechtsabteilung.

Relevante Datenschutzgrundsätze sind unter anderem:

| Grundsatz | Bedeutung für Paketmitschnitte |
|---|---|
| Rechtmäßigkeit und Transparenz | Verarbeitung benötigt eine tragfähige Grundlage |
| Zweckbindung | Daten nur für den festgelegten Diagnosezweck verwenden |
| Datenminimierung | nur erforderliche Kommunikation erfassen |
| Richtigkeit | Zeit, Messpunkt und Dateiintegrität dokumentieren |
| Speicherbegrenzung | Datei nicht länger als erforderlich aufbewahren |
| Integrität und Vertraulichkeit | Zugriff, Transport und Speicherung schützen |
| Rechenschaftspflicht | Zweck, Freigabe und Verarbeitung dokumentieren |

**Nicht ausreichend sind Aussagen wie:**

„Es ist nur für die Technik.“

„Die Verbindung ist verschlüsselt.“

„Wir schneiden nur kurz mit.“

Auch ein kurzer technischer Mitschnitt kann personenbezogene oder vertrauliche Daten enthalten.

## 4. Welche Freigaben sollten vor Beginn dokumentiert werden?

Ticketnummer:
Verantwortliche Person:
Auftraggebende Stelle:
Technischer Zweck:
Betroffene Systeme:
Betroffene Netzsegmente:
Betroffene Benutzergruppen:
Erlaubte Protokolle:
Ausgeschlossene Systeme und Daten:
Beginn:
Geplantes Ende:
Maximale Dateigröße:
Speicherort:
Auswertungsberechtigte Personen:
Vorgesehene Weitergabe:
Aufbewahrungsfrist:
Löschtermin:
Datenschutzprüfung erforderlich: Ja / Nein
Datenschutzprüfung erfolgt: Ja / Nein
Betriebsrat einzubeziehen: Ja / Nein / Nicht zutreffend
Informationssicherheit informiert: Ja / Nein
Freigabe erteilt durch:
Freigabezeitpunkt:

Bei einem akuten Sicherheitsvorfall können besondere Incident-Response-Regelungen gelten. Auch dann müssen Zuständigkeit, Beweissicherung und Zugriff nachvollziehbar dokumentiert werden.

## 5. Wo sollte ein Paketmitschnitt durchgeführt werden?

Der Messpunkt bestimmt, welcher Datenverkehr sichtbar ist.

| Messpunkt | Sichtbarer Datenverkehr | Typischer Zweck |
|---|---|---|
| betroffener Client | Kommunikation dieses Clients | lokales Clientproblem |
| Anwendungsserver | ein- und ausgehende Serverkommunikation | Backend- oder Dienstproblem |
| Reverse Proxy | Client- und Backendverbindungen des Proxys | HTTP- und TLS-Fehler eingrenzen |
| DNS-Server | DNS-Anfragen am Server | Auflösungsprobleme |
| Firewall | Verkehr an der jeweiligen Schnittstelle | Routing, NAT und Filterung |
| Switch-SPAN-Port | gespiegelter Verkehr ausgewählter Ports oder VLANs | segmentübergreifende Analyse |
| virtueller Switch | Verkehr virtueller Systeme | VM- und Hypervisoranalyse |
| Containerhost | Verkehr von Containern und Bridges | Containerkommunikation |

**Grundsatz**

> So nah wie möglich am vermuteten Problem mitschneiden.

Bei komplexen Problemen können zwei gleichzeitige Mitschnitte sinnvoll sein:

> Client-Mitschnitt + Server-Mitschnitt

Damit lässt sich beispielsweise prüfen:

- ob ein Paket den Client verlassen hat,
- ob es am Server angekommen ist,
- wo ein Verlust auftritt,
- ob NAT oder Firewall Adressen verändert,
- ob die Zeitstempel deutlich voneinander abweichen.

Vor einem Vergleich müssen die Systemuhren geprüft werden.

## 6. Wie wird die richtige Netzwerkschnittstelle bestimmt?

### Windows

```
[RO] Get-NetAdapter |
    Select-Object Name,
                  InterfaceDescription,
                  Status,
                  LinkSpeed,
                  MacAddress,
                  ifIndex
```

IP-Konfiguration zuordnen:

```
[RO] Get-NetIPConfiguration
```

Wireshark-Schnittstellen anzeigen:

```
[RO] dumpcap.exe -D
```

```
[RO] tshark.exe -D
```

### Linux

```
[RO] ip -brief address
```

```
[RO] ip route
```

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

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

### macOS

```
[RO] ifconfig
```

Standardroute und verwendetes Interface:

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

Wireshark-Schnittstellen:

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

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

**Typische Schnittstellen**

| Name | Mögliche Bedeutung |
|---|---|
| `Ethernet` | kabelgebundener Windows-Adapter |
| `Wi-Fi` | Windows-WLAN |
| `eth0`, `enp...`, `ens...` | Linux-Ethernet |
| `wlan0`, `wlp...` | Linux-WLAN |
| `en0`, `en1` | macOS-Netzwerkschnittstelle |
| `lo`, `lo0` | Loopback |
| `docker0` | Docker-Bridge |
| `br-...` | benutzerdefinierte Container-Bridge |
| `vEthernet (...)` | virtueller Windows-Adapter |
| `utun...` | macOS-Tunnel oder VPN |
| `any` | zusammengefasste Linux-Capture-Schnittstelle |

Die Namenskonvention allein reicht nicht. Die Schnittstelle muss über IP-Adresse, Route und tatsächliche Paketaktivität bestätigt werden.

## 7. Wie wird vor dem Mitschnitt ein Capture-Plan erstellt?

Fehler:

HTTPS-Aufruf an app.example.intern schlägt sporadisch fehl.

Betroffener Client:

CLIENT-023

Ziel:

192.0.2.20

Port:

TCP 443

Schnittstelle:

Ethernet / Index 4

Geplanter Zeitraum:

maximal 120 Sekunden

Auslöser:

Benutzer führt genau einen Anmeldeversuch aus.

Capture-Filter:

host 192.0.2.20 and tcp port 443

Datei:

TICKET-4711_CLIENT-023_20260731T094200+0200.pcapng

Maximale Dateigröße:

100 MB

Speicherort:

geschütztes Diagnoseverzeichnis

Nachbereitung:

Zeitfenster weiter reduzieren, Hash bilden, Zugriff begrenzen.

Löschtermin:

nach Abschluss gemäß Ticket und interner Richtlinie

Ein guter Capture-Plan verhindert unkontrollierte Daueraufzeichnungen und unnötig große Dateien.

## 8. Wie wird bereits bei der Aufnahme Datenminimierung umgesetzt?

Vier zentrale Begrenzungen:

1. richtige Schnittstelle,
2. enger Capture-Filter,
3. kurze Aufzeichnungsdauer,
4. begrenzte Paketlänge oder Dateigröße.

### Capture-Filter nach Host

```
host 192.0.2.20
```

### Nur Kommunikation zwischen zwei Systemen

```
host 192.0.2.10 and host 192.0.2.20
```

### Nur HTTPS zu einem Ziel

```
host 192.0.2.20 and tcp port 443
```

### Nur DNS zu einem bestimmten DNS-Server

```
host 192.0.2.53 and port 53
```

### Nur TCP zwischen zwei Systemen

```
tcp and host 192.0.2.10 and host 192.0.2.20
```

### Bestimmtes Subnetz ausschließen

```
host 192.0.2.20 and not net 198.51.100.0/24
```

### Wichtige Filterbegriffe

| Ausdruck | Bedeutung |
|---|---|
| `host 192.0.2.20` | Quelle oder Ziel ist dieser Host |
| `src host 192.0.2.20` | Host ist Quelle |
| `dst host 192.0.2.20` | Host ist Ziel |
| `net 192.0.2.0/24` | Verkehr eines Netzes |
| `port 53` | Quell- oder Zielport 53 |
| `src port 53` | Quellport 53 |
| `dst port 443` | Zielport 443 |
| `tcp` | nur TCP |
| `udp` | nur UDP |
| `icmp` | nur ICMP |
| `and` | beide Bedingungen müssen zutreffen |
| `or` | mindestens eine Bedingung trifft zu |
| `not` | Bedingung ausschließen |

Capture-Filter verwenden die Syntax von libpcap/BPF. Sie unterscheiden sich von Wireshark-Display-Filtern.

## 9. Was ist der Unterschied zwischen Capture- und Display-Filter?

| Filterart | Zeitpunkt | Wirkung |
|---|---|---|
| Capture-Filter | während der Aufnahme | nicht passende Pakete werden nicht gespeichert |
| Display-Filter | nach beziehungsweise während der Anzeige | Pakete bleiben in der Datei, werden aber ausgeblendet |

### Capture-Filter

```
host 192.0.2.20 and tcp port 443
```

### Entsprechender Display-Filter

```
ip.addr == 192.0.2.20 && tcp.port == 443
```

Ein Display-Filter reduziert nicht den sensiblen Inhalt der gespeicherten Originaldatei.

```
Ausgeblendet ≠ entfernt
```

Vor einer Weitergabe muss eine neue, tatsächlich reduzierte Datei erzeugt und anschließend erneut geprüft werden.

## 10. Wie wird mit dumpcap eine zeitlich begrenzte Aufnahme erstellt?

`dumpcap` ist die spezialisierte Capture-Komponente von Wireshark. Die grafische Wireshark-Anwendung muss dadurch nicht mit erhöhten Rechten ausgeführt werden.

### Schnittstellen auflisten

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

### 60 Sekunden auf Schnittstelle 1 mitschneiden

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -a duration:60 \
  -w capture.pcapng
```

### Gezielter HTTPS-Mitschnitt

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -f "host 192.0.2.20 and tcp port 443" \
  -a duration:60 \
  -w capture.pcapng
```

### Nach 10.000 Paketen stoppen

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -f "host 192.0.2.20" \
  -c 10000 \
  -w capture.pcapng
```

### Nach einer Dateigröße von ungefähr 100.000 kB stoppen

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -f "host 192.0.2.20" \
  -a filesize:100000 \
  -w capture.pcapng
```

Bei `dumpcap` wird `filesize` in Kilobyte zu jeweils 1.000 Byte angegeben.

### Wichtige Optionen

| Option | Bedeutung |
|---|---|
| `-D` | Schnittstellen auflisten |
| `-i` | Schnittstelle auswählen |
| `-f` | Capture-Filter |
| `-a duration:60` | nach 60 Sekunden stoppen |
| `-a filesize:100000` | bei ungefähr 100 MB stoppen |
| `-c 10000` | nach 10.000 Paketen stoppen |
| `-s 96` | maximal 96 Byte jedes Pakets speichern |
| `-w` | Ausgabedatei |
| `-p` | Promiscuous Mode deaktivieren |
| `-q` | reduzierte Statusausgabe |

Der tatsächliche Schnittstellenname oder die Nummer muss vorher mit `dumpcap -D` ermittelt werden.

## 11. Wie wird mit tcpdump eine begrenzte Aufnahme erstellt?

`tcpdump` ist unter Linux und macOS häufig verfügbar. Unter Windows werden üblicherweise Wireshark, Dumpcap oder TShark mit Npcap verwendet.

### 60 Sekunden gezielt aufzeichnen

Linux:

```
[TEST][PRIV][FILE][SENS] sudo timeout 60 \
  tcpdump -i eth0 \
  -nn \
  -w capture.pcap \
  'host 192.0.2.20 and tcp port 443'
```

`timeout` gehört zum GNU-Coreutils-Umfeld und ist unter macOS nicht standardmäßig in derselben Form vorhanden.

### Nach 10.000 Paketen stoppen

Linux und macOS:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump \
  -i en0 \
  -nn \
  -c 10000 \
  -w capture.pcap \
  'host 192.0.2.20 and tcp port 443'
```

### Paketlänge auf 96 Byte begrenzen

```
[TEST][PRIV][FILE][SENS] sudo tcpdump \
  -i en0 \
  -nn \
  -s 96 \
  -c 10000 \
  -w capture-truncated.pcap \
  'host 192.0.2.20 and tcp port 443'
```

### Wichtige Optionen

| Option | Bedeutung |
|---|---|
| `-i en0` | Aufnahmeschnittstelle |
| `-nn` | keine Hostnamen- und Portnamensauflösung |
| `-c 10000` | nach 10.000 Paketen stoppen |
| `-s 96` | Snapshot-Länge 96 Byte |

| Option | Bedeutung |
| --- | --- |
| `-w DATEI` | Rohpakete in Datei schreiben |
| `-r DATEI` | vorhandene Datei lesen |
| Filter am Ende | Capture-Filter |

> Eine geringe Snapshot-Länge reduziert den gespeicherten Paketinhalt, kann aber wichtige Protokollinformationen abschneiden. Außerdem bleiben Netzwerkheader, Adressen und möglicherweise Teile der Nutzdaten erhalten. Trunkierung ist keine vollständige Anonymisierung.

## 12. Wie wird eine Ringpuffer-Aufzeichnung sicher begrenzt?

Ein Ringpuffer schreibt mehrere Dateien und überschreibt nach Erreichen der festgelegten Anzahl die älteste Datei. Dadurch wird der maximale Speicherverbrauch begrenzt.

**Dumpcap: fünf Dateien mit jeweils 60 Sekunden**

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -f "host 192.0.2.20 and tcp port 443" \
  -b duration:60 \
  -b files:5 \
  -w capture.pcapng
```

**Dumpcap: fünf Dateien mit jeweils ungefähr 50 MB**

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -f "host 192.0.2.20 and tcp port 443" \
  -b filesize:50000 \
  -b files:5 \
  -w capture.pcapng
```

**tcpdump: fünf Dateien mit ungefähr 50 MB**

```
[TEST][PRIV][FILE][SENS] sudo tcpdump \
  -i eth0 \
  -nn \
  -C 50 \
  -W 5 \
  -w capture.pcap \
  'host 192.0.2.20 and tcp port 443'
```

Bei `tcpdump` können Einheit und Verhalten einzelner Rotationsoptionen von der verwendeten Implementierung und Version abhängen. Vor dem Einsatz muss die lokale Dokumentation geprüft werden:

```
[RO] man tcpdump
```

**Wichtiger Hinweis**

Ein Ringpuffer verhindert unbegrenztes Dateiwachstum, löscht aber ältere Aufzeichnungen automatisch durch Überschreiben. Er ist deshalb nicht geeignet, wenn alle Daten unverändert als Beweismittel erhalten bleiben müssen.

## 13. Wie wird der Promiscuous Mode bewertet?

Im Promiscuous Mode nimmt eine Netzwerkkarte zusätzlich Frames entgegen, die nicht unmittelbar an ihre eigene MAC-Adresse adressiert sind.

```
Promiscuous Mode ≠ automatisch gesamter Netzwerkverkehr
```

In einem geswitchten Netzwerk sieht ein Client normalerweise weiterhin nur:

- eigenen Unicast-Verkehr,
- Broadcasts,
- relevante Multicasts,
- vom Switch an diesen Port weitergeleiteten Verkehr.

Für zusätzlichen Verkehr ist häufig ein korrekt konfigurierter SPAN- beziehungsweise Mirror-Port oder ein anderer geeigneter Messpunkt erforderlich.

**Promiscuous Mode mit dumpcap deaktivieren**

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i 1 \
  -p \
  -f "host 192.0.2.20" \
  -a duration:60 \
  -w capture.pcapng
```

Wenn nur lokaler Hostverkehr benötigt wird, kann die Deaktivierung des Promiscuous Mode die Erfassung unbeabsichtigten Verkehrs reduzieren.

## 14. Was ist bei WLAN-Mitschnitten und Monitor Mode zu beachten?

Der WLAN-Monitor-Mode kann rohe 802.11-Frames eines Funkkanals erfassen. Unterstützung und Verhalten hängen ab von:

- Betriebssystem,
- WLAN-Chipsatz,
- Treiber,
- Kanal,
- Frequenzband,
- Verschlüsselung,
- aktueller Verbindung.

Das Aktivieren des Monitor Mode kann die bestehende WLAN-Verbindung unterbrechen:

```
[CHANGE][DISRUPT]
```

Zusätzlich können Kommunikationsdaten anderer Geräte im Funkbereich erfasst werden. Ein WLAN-Mitschnitt benötigt daher eine besonders klare technische und organisatorische Abgrenzung.

Für eine gewöhnliche Client-Fehleranalyse ist häufig ein Mitschnitt am regulären Netzwerkinterface ohne Monitor Mode ausreichend.

## 15. Wie wird eine laufende Aufnahme überwacht?

Während der Aufnahme sind folgende Werte zu kontrollieren:

- Dateigröße,

- verbleibender Speicherplatz,
- Paketanzahl,
- verworfene Pakete,
- CPU-Auslastung,
- Schreibgeschwindigkeit,
- Aufnahmezeit,
- korrekte Schnittstelle,
- korrekter Filter,
- Zeitpunkt der Fehlerreproduktion.

**Speicherplatz prüfen**

Windows:

```
[RO] Get-Volume
```

Linux:

```
[RO] df -h
```

macOS:

```
[RO] df -h
```

**Dateigröße beobachten**

Windows:

```
[RO] Get-Item '.\capture.pcapng' |
    Select-Object FullName, Length, LastWriteTime
```

Linux und macOS:

```
[RO] ls -lh capture.pcapng
```

Wenn Capture-Pakete verworfen werden, ist die Aufzeichnung möglicherweise unvollständig. Mögliche Ursachen:

- Capture-Puffer zu klein,
- Datenträger zu langsam,

- zu hohe Paketrate,
- System zu stark ausgelastet,
- zu umfangreiche Live-Anzeige,
- ungeeigneter Messpunkt.

Ein größerer Capture-Puffer kann helfen, benötigt aber zusätzlichen Arbeitsspeicher und ersetzt keine Bewertung der Systemkapazität.

## 16. Wie wird ein Mitschnitt unmittelbar nach der Aufnahme geprüft?

### Dateiinformationen anzeigen

Windows, Linux und macOS:

```
[RO][SENS] capinfos capture.pcapng
```

Wichtige Informationen:

- Dateiformat,
- Dateigröße,
- Anzahl der Pakete,
- erstes und letztes Paket,
- Aufzeichnungsdauer,
- Schnittstelle,
- Kapselung,
- Datenrate,
- gespeicherte Paketlängen,
- vorhandene Kommentare oder Geheimnisse.

### Kurze Protokollübersicht

```
[RO][SENS] tshark -r capture.pcapng -q -z io,phs
```

### Kommunikationsbeziehungen

```
[RO][SENS] tshark -r capture.pcapng -q -z endpoints,ip
```

```
[RO][SENS] tshark -r capture.pcapng -q -z conv,tcp
```

### Prüffragen

1. Enthält die Datei den erwarteten Zeitraum?
2. Ist die richtige Schnittstelle erfasst?
3. Ist der Zielverkehr enthalten?
4. Wurde unbeabsichtigter Fremdverkehr gespeichert?
5. Sind unverschlüsselte Inhalte sichtbar?
6. Enthält die Datei eingebettete Entschlüsselungsgeheimnisse?
7. Sind Kommentare oder Metadaten vorhanden?
8. Ist eine weitere Reduktion erforderlich?
9. Muss die Originaldatei erhalten bleiben?
10. Wer darf die Datei öffnen?

## 17. Wie wird aus einer großen Aufzeichnung eine zeitlich begrenzte Arbeitskopie erstellt?

Mit `editcap` kann ein Zeitbereich aus einer vorhandenen Datei in eine neue Datei geschrieben werden.

### Zeitbereich mit UTC-Offset auswählen

```
[RO][FILE][SENS] editcap \
  -A "2026-07-31T09:42:00+02:00" \
  -B "2026-07-31T09:44:00+02:00" \
  original.pcapng \
  arbeitskopie.pcapng
```

`-A` nimmt Pakete ab dem angegebenen Zeitpunkt auf. `-B` begrenzt auf Pakete vor dem angegebenen Endzeitpunkt.

### Bestimmte Paketnummern übernehmen

Nur Pakete 200 bis 750 schreiben:

```
[RO][FILE][SENS] editcap \
  -r \
  original.pcapng \
  arbeitskopie.pcapng \
  200-750
```

### Erste 500 Pakete übernehmen

```
[RO][FILE][SENS] editcap \
  -r \
  original.pcapng \
  erste-500.pcapng \
  1-500
```

Die neue Datei muss anschließend erneut mit Wireshark, TShark oder Capinfos geprüft werden. Eine erfolgreiche Programmausführung beweist nicht, dass alle unerwünschten Daten entfernt wurden.

## 18. Wie wird eine Datei nach einem Display-Filter exportiert?

Mit TShark kann aus einer vorhandenen Aufzeichnung eine gefilterte Datei erzeugt werden.

### Nur Verkehr eines bestimmten Hosts exportieren

```
[RO][FILE][SENS] tshark \
  -r original.pcapng \
  -Y "ip.addr == 192.0.2.20" \
  -w gefiltert.pcapng
```

### Nur TCP-Port 443 zu einem bestimmten Host

```
[RO][FILE][SENS] tshark \
  -r original.pcapng \
  -Y "ip.addr == 192.0.2.20 && tcp.port == 443" \
  -w gefiltert.pcapng
```

### Nur ein bestimmtes TCP-Gespräch

Zunächst Streamnummer in Wireshark oder TShark bestimmen. Anschließend beispielsweise:

```
[RO][FILE][SENS] tshark \
  -r original.pcapng \
  -Y "tcp.stream == 4" \
```

```
  -w tcp-stream-4.pcapng
```

**Wichtig**

- Ein Export nach Display-Filter erzeugt eine neue reduzierte Datei.
- Das Original bleibt unverändert.
- Nicht angezeigte Pakete werden beim Export nicht übernommen.
- Die übernommenen Pakete enthalten weiterhin ihre vollständigen gespeicherten Header und Nutzdaten.
- Die gefilterte Datei ist deshalb nicht automatisch anonymisiert.

## 19. Wie werden eingebettete Entschlüsselungsgeheimnisse und Kommentare entfernt?

PCAPNG-Dateien können zusätzliche Metadaten enthalten, darunter:

- Capture-Kommentare,
- Paketkommentare,
- Schnittstelleninformationen,
- Decryption Secrets Blocks.

Vor einer Weitergabe muss geprüft werden, ob solche Informationen vorhanden sind.

**Eingebettete Entschlüsselungsgeheimnisse entfernen**

```
[RO][FILE][SENS] editcap \
  --discard-all-secrets \
  original.pcapng \
  ohne-secrets.pcapng
```

**Capture- und Paketkommentare entfernen**

```
[RO][FILE][SENS] editcap \
  --discard-capture-comment \
  --discard-packet-comments \
  original.pcapng \
  ohne-kommentare.pcapng
```

**Kombiniert**

```
[RO][FILE][SENS] editcap \
  --discard-all-secrets \
  --discard-capture-comment \
  --discard-packet-comments \
  original.pcapng \
  bereinigt.pcapng
```

Danach muss die Ausgabedatei erneut geprüft werden:

```
[RO][SENS] capinfos bereinigt.pcapng
```

Das Entfernen von Geheimnisblöcken und Kommentaren entfernt nicht automatisch IP-Adressen, MAC-Adressen, Hostnamen oder Nutzdaten.

## 20. Warum ist Trunkierung keine vollständige Anonymisierung?

Mit `editcap -s` kann die gespeicherte Länge jedes Pakets begrenzt werden.

**Auf 96 Byte kürzen**

```
[RO][FILE][SENS] editcap \
  -s 96 \
  original.pcapng \
  gekuerzt.pcapng
```

Dadurch können Teile der Nutzdaten entfernt werden. Abhängig von den vorhandenen Protokollheadern können aber weiterhin enthalten sein:

- MAC-Adressen,
- VLAN-Informationen,
- IP-Adressen,
- Ports,
- TCP-Flags,
- DNS-Inhalte,
- TLS-Informationen,
- Teile der Nutzdaten.

Eine feste Snapshot-Länge kann außerdem die technische Analyse beeinträchtigen.

```
Gekürzt ≠ anonymisiert
```

Eine belastbare Anonymisierung erfordert:

- Definition der zu schützenden Felder,
- geeignetes Werkzeug und Verfahren,
- Prüfung der resultierenden Datei,
- Berücksichtigung eingebetteter Metadaten,
- Berücksichtigung rekonstruierbarer Zusammenhänge.

Wireshark und Editcap bieten keine allgemeine Ein-Klick-Anonymisierung, die für jedes Protokoll zuverlässig sämtliche personenbezogenen oder vertraulichen Inhalte entfernt.

## 21. Wie werden Hashwerte zur Integritätsprüfung erstellt?

Ein kryptografischer Hash dokumentiert den Zustand einer Datei zu einem bestimmten Zeitpunkt. Wird die Datei verändert, ändert sich mit sehr hoher Wahrscheinlichkeit auch der Hash.

**Windows**

```
[RO] Get-FileHash `
    'C:\Diagnose\original.pcapng' `
    -Algorithm SHA256
```

Hash in Datei dokumentieren:

```
[RO][FILE] Get-FileHash `
    'C:\Diagnose\original.pcapng' `
    -Algorithm SHA256 |
    Format-List |
    Out-File 'C:\Diagnose\original.pcapng.sha256.txt'
```

**Linux**

```
[RO] sha256sum original.pcapng
```

```
[RO][FILE] sha256sum original.pcapng \
  > original.pcapng.sha256
```

**macOS**

```
[RO] shasum -a 256 original.pcapng
```

```
[RO][FILE] shasum -a 256 original.pcapng \
  > original.pcapng.sha256
```

**Erneut prüfen**

Linux:

```
[RO] sha256sum -c original.pcapng.sha256
```

macOS:

```
[RO] shasum -a 256 -c original.pcapng.sha256
```

Ein Hash beweist nicht automatisch, wer die Datei erstellt hat oder ob der ursprüngliche Mitschnitt vollständig war. Er hilft aber festzustellen, ob sich genau diese Datei nach der Hashbildung verändert hat.

## 22. Wie werden Original und Arbeitskopie getrennt?

Empfohlene Struktur:

```
TICKET-4711/
├── original/
│   ├── TICKET-4711_CLIENT-023_original.pcapng
│   └── TICKET-4711_CLIENT-023_original.pcapng.sha256
├── working/
│   ├── TICKET-4711_zeitfenster.pcapng
│   └── TICKET-4711_tcp-stream-4.pcapng
├── export/
```

```
│   └── TICKET-4711_freigegebene-kopie.pcapng
└── dokumentation/
    └── TICKET-4711_capture-protokoll.txt
```

**Regeln**

- Original nach der Sicherung nicht mehr bearbeiten.
- Hash des Originals unmittelbar nach der Aufnahme erstellen.
- Analysen auf einer Arbeitskopie durchführen.
- Weitergabekopie getrennt erzeugen.
- Jede Bearbeitung dokumentieren.
- Weitergabekopie vor Übergabe erneut kontrollieren.
- Hashwerte von Original und Export getrennt dokumentieren.

Das Original enthält möglicherweise mehr sensible Daten als die freigegebene Weitergabekopie und benötigt entsprechend strengere Zugriffsrechte.

## 23. Wie werden Dateizugriffe technisch eingeschränkt?

**Windows-Beispiel**

Vererbung entfernen:

```
[CHANGE][PRIV] icacls "C:\Diagnose\TICKET-4711" /inheritance:r
```

Einem ausdrücklich festgelegten Konto Zugriff gewähren:

```
[CHANGE][PRIV] icacls "C:\Diagnose\TICKET-4711" /grant:r "DOMÄNE\Diagnosekonto:(OI)(CI)F"
```

Vorher müssen Konto, gewünschte Rechte und bestehende Berechtigungen genau geprüft werden. Ein falscher `icacls`-Befehl kann berechtigte Personen aussperren.

Aktuelle Berechtigungen anzeigen:

```
[RO] icacls "C:\Diagnose\TICKET-4711"
```

**Linux und macOS**

Verzeichnis nur für den Eigentümer zugänglich machen:

```
[CHANGE] chmod 700 /pfad/TICKET-4711
```

Datei nur für den Eigentümer les- und schreibbar machen:

```
[CHANGE] chmod 600 /pfad/TICKET-4711/original.pcapng
```

Berechtigungen prüfen:

```
[RO] ls -ld /pfad/TICKET-4711
```

```
[RO] ls -l /pfad/TICKET-4711
```

Dateiberechtigungen ersetzen keine Verschlüsselung, keine sichere Übertragung und keine organisatorische Zugriffskontrolle.

## 24. Wie wird ein Paketmitschnitt sicher weitergegeben?

Vor jeder Weitergabe:

1. Empfänger und Zweck bestätigen.
2. Vertragliche und datenschutzrechtliche Zulässigkeit prüfen.
3. Nur den erforderlichen Zeit- und Datenbereich exportieren.
4. Unnötige Pakete entfernen.
5. Kommentare und eingebettete Geheimnisse entfernen.
6. Datei auf unverschlüsselte Inhalte prüfen.
7. Freigegebene Kopie getrennt speichern.
8. Hash der freigegebenen Datei berechnen.
9. Sicheren Übertragungsweg verwenden.
10. Kennwort oder Schlüssel über einen getrennten Kanal übermitteln.
11. Übergabe im Ticket dokumentieren.
12. Löschung beim Empfänger vereinbaren und dokumentieren.

### Nicht geeignete Übertragungswege

- öffentlicher Dateilink ohne Zugriffsschutz,
- unverschlüsselte E-Mail,
- privater Messenger ohne Freigabe,
- öffentliches Cloudlaufwerk,
- frei zugänglicher Webserver,

- unkontrollierter USB-Datenträger.

Ein vom Hersteller bereitgestelltes Supportportal ist nicht automatisch für beliebige vertrauliche Daten freigegeben. Vertragslage, Standort, Auftragsverarbeitung und interne Vorgaben müssen geprüft werden.

## 25. Was ist bei TLS-Entschlüsselung zu beachten?

TLS-Entschlüsselung kann Inhalte sichtbar machen, die gerade durch Verschlüsselung geschützt werden sollen.

Mögliche Inhalte:

- Zugangsdaten,
- Sitzungscookies,
- API-Token,
- personenbezogene Daten,
- Formulardaten,
- Dateiinhalte,
- interne Anwendungsinformationen.

TLS-Key-Log-Dateien und private Schlüssel sind besonders schützenswert:

> [SENS] Eine TLS-Key-Log-Datei kann die Entschlüsselung aufgezeichneter Sitzungen ermöglichen.

**Regeln**

- nur mit ausdrücklicher Freigabe,
- nur für den erforderlichen Prozess oder Testbenutzer,
- möglichst in kontrollierter Testumgebung,
- Key-Log-Datei getrennt schützen,
- nicht gemeinsam mit der Aufzeichnung unkontrolliert versenden,
- nach Abschluss kontrolliert löschen,
- niemals produktive private Server-Schlüssel exportieren, wenn dies nicht ausdrücklich erforderlich und freigegeben ist.

PCAPNG-Dateien können Entschlüsselungsgeheimnisse eingebettet enthalten. Vor einer Weitergabe muss dies ausdrücklich geprüft werden.

## 26. Welche Fehler treten bei Paketmitschnitten häufig auf?

| Fehler | Folge | Besseres Vorgehen |
|---|---|---|
| falsche Schnittstelle | relevanter Verkehr fehlt | IP-Konfiguration und Route vorher prüfen |
| Aufnahme am falschen Messpunkt | Ursache bleibt unsichtbar | Kommunikationsweg zeichnen |
| kein Capture-Filter | unnötig viele sensible Daten | Zielhost und Port begrenzen |
| nur Display-Filter verwendet | Fremdverkehr bleibt in Datei | reduzierte Datei exportieren |
| Aufnahme läuft unbegrenzt | Speicherplatz und Datenschutzproblem | Dauer, Größe oder Ringpuffer begrenzen |
| Fehler nicht reproduziert | relevanter Vorgang fehlt | exakten Reproduktionszeitpunkt notieren |
| Namensauflösung aktiv | Anzeige wird verfälscht oder zusätzliche DNS-Last entsteht | bei Analyse gegebenenfalls numerische Anzeige verwenden |
| Zeitabweichung ignoriert | Client- und Servermitschnitt passen nicht zusammen | Uhren und Zeitzonen prüfen |
| Snapshot-Länge zu klein | benötigte Protokolldaten fehlen | Anforderung vor Aufnahme bestimmen |
| Snapshot-Länge unbegrenzt | unnötige Nutzdaten werden gespeichert | erforderliche Länge bewusst festlegen |
| Original direkt bearbeitet | Nachvollziehbarkeit geht verloren | Original und Arbeitskopie trennen |
| Datei unverschlüsselt versendet | Datenabfluss möglich | freigegebenen sicheren Kanal verwenden |
| PCAP öffentlich hochgeladen | vertrauliche Daten offengelegt | interne oder freigegebene Analyse |
| Datei nur umbenannt | Inhalt bleibt vollständig erhalten | echte Reduktion und Prüfung durchführen |
| Hash erst nach Bearbeitung erstellt | ursprünglicher Zustand nicht belegt | Hash unmittelbar nach Sicherung bilden |
| Wireshark als Root gestartet | unnötig große Angriffsfläche | Capture-Funktion auf Dumpcap beschränken |

## 27. Wie läuft ein sicherer Paketmitschnitt vollständig ab?

| Phase | Vorgehen |
|---|---|
| 1. Auftrag klären | technisches Problem und Zweck dokumentieren |
| 2. Freigabe einholen | Zuständigkeit, Datenschutz und Sicherheit klären |
| 3. Kommunikationsweg zeichnen | Client, Netzwerkkomponenten und Server bestimmen |
| 4. Messpunkt wählen | so nah wie möglich am vermuteten Problem |

| Phase | Vorgehen |
|---|---|
| 5. Schnittstelle prüfen | Interface, IP-Adresse und Route bestätigen |
| 6. Filter festlegen | Host, Port und Protokoll begrenzen |
| 7. Umfang begrenzen | Dauer, Paketanzahl, Dateigröße und Snapshot-Länge |
| 8. Speicher schützen | Zugriffsrechte und freien Speicherplatz prüfen |
| 9. Aufnahme starten | Startzeit und Werkzeugversion dokumentieren |
| 10. Fehler reproduzieren | genau definierte Aktion ausführen |
| 11. Aufnahme stoppen | unmittelbar nach dem relevanten Vorgang |
| 12. Datei prüfen | Zeitraum, Pakete, Fremdverkehr und Geheimnisse |
| 13. Original sichern | Hash bilden und Original unverändert ablegen |
| 14. Arbeitskopie erzeugen | Zeitfenster und relevante Kommunikation reduzieren |
| 15. Technisch analysieren | Protokollablauf und Fehlerhypothese untersuchen |
| 16. Weitergabe vorbereiten | Daten minimieren und Freigabe einholen |
| 17. Ergebnis dokumentieren | Ursache, Belege und Grenzen festhalten |
| 18. Daten löschen | Aufbewahrungsfrist und Löschbestätigung beachten |

## 28. Kompakte Befehlsübersicht

| Aufgabe | Befehl | Kennzeichnung |
|---|---|---|
| Capture-Schnittstellen anzeigen | `dumpcap -D` | [RO] |
| tcpdump-Schnittstellen anzeigen | `tcpdump -D` | [RO] |
| 60 Sekunden aufzeichnen | `dumpcap -i 1 -a duration:60 -w capture.pcapng` | [TEST][PRIV][FILE][SENS] |
| gefiltert aufzeichnen | `dumpcap -i 1 -f "host 192.0.2.20 and tcp port 443" -a duration:60 -w capture.pcapng` | [TEST][PRIV][FILE][SENS] |
| nach 10.000 Paketen stoppen | `dumpcap -i 1 -c 10000 -w capture.pcapng` | [TEST][PRIV][FILE][SENS] |
| Paketlänge begrenzen | `dumpcap -i 1 -s 96 -a duration:60 -w capture.pcapng` | [TEST][PRIV][FILE][SENS] |
| Ringpuffer | `dumpcap -i 1 -b duration:60 -b files:5 -w capture.pcapng` | [TEST][PRIV][FILE][SENS] |
| Dateiinformationen | `capinfos capture.pcapng` | [RO][SENS] |
| Protokollhierarchie | `tshark -r capture.pcapng -q -z io,phs` | [RO][SENS] |
| IP-Endpunkte | `tshark -r capture.pcapng -q -z endpoints,ip` | [RO][SENS] |

| Aufgabe | Befehl | Kennzeichnung |
|---|---|---|
| TCP-Gespräche | `tshark -r capture.pcapng -q -z conv,tcp` | [RO][SENS] |
| Zeitbereich exportieren | `editcap -A "START" -B "ENDE" original.pcapng arbeitskopie.pcapng` | [RO][FILE][SENS] |
| Paketbereich exportieren | `editcap -r original.pcapng auszug.pcapng 200-750` | [RO][FILE][SENS] |
| Pakete nach Filter exportieren | `tshark -r original.pcapng -Y "ip.addr == 192.0.2.20" -w gefiltert.pcapng` | [RO][FILE][SENS] |
| Geheimnisblöcke entfernen | `editcap --discard-all-secrets original.pcapng bereinigt.pcapng` | [RO][FILE][SENS] |
| Kommentare entfernen | `editcap --discard-capture-comment --discard-packet-comments original.pcapng bereinigt.pcapng` | [RO][FILE][SENS] |
| Paketdaten kürzen | `editcap -s 96 original.pcapng gekuerzt.pcapng` | [RO][FILE][SENS] |
| SHA-256 unter Windows | `Get-FileHash .\capture.pcapng -Algorithm SHA256` | [RO] |
| SHA-256 unter Linux | `sha256sum capture.pcapng` | [RO] |
| SHA-256 unter macOS | `shasum -a 256 capture.pcapng` | [RO] |

## 29. Dokumentationsvorlage für einen Paketmitschnitt

Ticketnummer:

Auftraggebende Stelle:

Ausführende Person:

Freigabe erteilt durch:

Freigabezeitpunkt:

Datenschutzprüfung:

Informationssicherheit informiert:

Betriebsrat einbezogen oder nicht erforderlich:

Technischer Zweck:

Fehlerbeschreibung:

Betroffene Systeme:

Betroffene Benutzergruppe:

Ausgeschlossene Systeme:

Erlaubter Datenumfang:

Capture-System:

Betriebssystem:

Capture-Werkzeug:

Werkzeugversion:

Capture-Schnittstelle:

IP-Adresse der Schnittstelle:

Capture-Position:

Promiscuous Mode: Ja / Nein

Monitor Mode: Ja / Nein

Capture-Filter:

Snapshot-Länge:

Maximale Dauer:

Maximale Paketanzahl:

Maximale Dateigröße:

Ringpuffer: Ja / Nein

Anzahl Ringpufferdateien:

Startzeit:

Endzeit:

Zeitzone:

Zeitpunkt der Fehlerreproduktion:

Durchgeführte Aktion:

Originaldatei:

Dateigröße:

Paketanzahl:

Erstes Paket:

Letztes Paket:

SHA-256 des Originals:

Speicherort des Originals:

Zugriffsberechtigte Personen:

Arbeitskopie:

Verwendeter Zeitfilter:

Verwendeter Display-Filter:

Entfernte Kommentare:

Entfernte Entschlüsselungsgeheimnisse:

Snapshot-Länge nach Bearbeitung:

SHA-256 der Arbeitskopie:

Ermittelte Beobachtungen:

1.

2.

3.

Arbeitshypothese:

Beleg:

Gegenprüfung:

Ermittelte Ursache:

Weitergabe erforderlich: Ja / Nein

Empfänger:

Rechts- und Datenschutzprüfung:

Weitergabedatei:

SHA-256 der Weitergabedatei:

Übertragungsweg:

Übergabezeitpunkt:

Löschvereinbarung mit Empfänger:

Interne Aufbewahrungsfrist:

Geplanter Löschtermin:

Tatsächlicher Löschzeitpunkt:

Löschung bestätigt durch:

## 30. Offizielle Quellen und weiterführende Dokumentation

### Wireshark

- Wireshark User’s Guide
- Wireshark Capture Options
- Wireshark Capture Filters
- Wireshark Display Filters
- dumpcap – offizielle Manual Page
- tshark – offizielle Manual Page
- editcap – offizielle Manual Page
- capinfos – offizielle Manual Page

- Wireshark Manual Pages

**tcpdump und libpcap**

- tcpdump – offizielle Dokumentation
- pcap-filter – offizielle Filtersyntax

**Datenschutzrecht**

- Datenschutz-Grundverordnung – EUR-Lex
- Bundesbeauftragte für den Datenschutz und die Informationsfreiheit
- Datenschutzkonferenz des Bundes und der Länder

> Welche rechtliche Grundlage, Beteiligung oder Dokumentation im konkreten Unternehmen erforderlich ist, muss durch die zuständigen Datenschutz-, Sicherheits- und Rechtsstellen entschieden werden. Die technische Möglichkeit eines Paketmitschnitts stellt keine automatische Berechtigung zur Durchführung dar.

---