2.5 Wireshark und TShark – Netzwerkverkehr aufzeichnen und analysieren

Ziel dieser Seite

Wireshark und TShark ermöglichen die detaillierte Untersuchung des Netzwerkverkehrs. Sie helfen unter anderem bei der Analyse von:

- Verbindungsabbrüchen;
- langsamen Netzwerkverbindungen;
- DNS-Problemen;
- TCP-Verbindungsfehlern;
- Paketverlusten und Übertragungswiederholungen;
- DHCP-Problemen;
- TLS-Verbindungsabbrüchen;
- fehlerhaften Anwendungen und Protokollen;
- Kommunikationsproblemen zwischen Clients und Servern.

Ein Paketmitschnitt sollte erst dann begonnen werden, wenn die betroffenen Systeme, der ungefähre Fehlerzeitpunkt und der erwartete Kommunikationsweg bekannt sind.

> **Wichtig:** Paketmitschnitte dürfen ausschließlich in Netzwerken und auf Systemen durchgeführt werden, für die eine entsprechende Berechtigung vorliegt. Aufzeichnungen können Benutzernamen, IP-Adressen, DNS-Anfragen, Sitzungsinformationen und unverschlüsselte Anwendungsdaten enthalten.

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]`	Die Ausgabe oder Datei kann sensible Informationen enthalten
`[DISRUPT]`	Der Vorgang kann Systeme oder Netzwerkverbindungen beeinflussen

---

1. Was ist der Unterschied zwischen Wireshark, TShark und Dumpcap?

Werkzeuge und Aufgaben anzeigen

Werkzeug	Aufgabe	Typischer Einsatz
Wireshark	Grafische Paketaufzeichnung und Analyse	Interaktive Fehlersuche am Arbeitsplatz
TShark	Kommandozeilenversion der Wireshark-Analyse	Server, Skripte und automatisierte Auswertungen
Dumpcap	Spezialisierte Paketaufzeichnung	Ressourcenarme oder länger laufende Mitschnitte
Capinfos	Zeigt Eigenschaften einer Mitschnittdatei	Kontrolle von Dateiformat, Zeitraum und Paketanzahl
Mergecap	Führt mehrere Mitschnittdateien zusammen	Gemeinsame Analyse mehrerer Aufzeichnungen
Editcap	Bearbeitet oder verkleinert Mitschnittdateien	Zeitbereiche oder einzelne Pakete extrahieren

Wireshark und TShark verwenden dieselben Protokoll-Dissektoren und dieselbe Display-Filter-Sprache. Die eigentliche Paketaufzeichnung wird üblicherweise von Dumpcap durchgeführt.

Grundlegende Versionsprüfung

Aufgabe	Befehl
Wireshark-Version anzeigen	[RO] wireshark --version
TShark-Version anzeigen	[RO] tshark --version
Dumpcap-Version anzeigen	[RO] dumpcap --version
Capinfos-Version anzeigen	[RO] capinfos --version

---

2. Welche Voraussetzungen müssen vor einem Paketmitschnitt geprüft werden?

Vorbedingungen und Berechtigungen anzeigen

Vor dem Mitschnitt müssen mindestens folgende Punkte geklärt werden:

Prüffrage	Bedeutung
Welches System ist betroffen?	Bestimmt den geeigneten Aufzeichnungspunkt
Welche Gegenstelle wird angesprochen?	Ermöglicht einen gezielten Host- oder Portfilter

Prüffrage	Bedeutung
Wann tritt der Fehler auf?	Verhindert unnötig lange Aufzeichnungen
Welches Protokoll wird erwartet?	Erleichtert die spätere Analyse
Welche Schnittstelle transportiert den Verkehr?	Verhindert einen Mitschnitt auf dem falschen Adapter
Ist der Mitschnitt autorisiert?	Schützt personenbezogene und betriebliche Daten
Gibt es ausreichend Speicherplatz?	Verhindert eine volle Festplatte
Sind die Systemuhren synchronisiert?	Ermöglicht den Vergleich mit Server- und Anwendungslogs

Betriebssystemspezifische Hinweise

Betriebssystem	Typische Voraussetzung
Windows	Für Live-Mitschnitte wird normalerweise Npcap benötigt
Linux	Der Benutzer benötigt Zugriff auf Dumpcap beziehungsweise die Capture-Gruppe; alternativ sind erhöhte Rechte erforderlich
macOS	Die Wireshark-Installation richtet üblicherweise die notwendigen BPF-Berechtigungen ein
Remote-Server	Häufig wird TShark oder Dumpcap ohne grafische Oberfläche verwendet

Wireshark oder TShark sollten nicht dauerhaft als Administrator beziehungsweise `root` ausgeführt werden. Besser ist eine gezielte Berechtigung ausschließlich für die Paketaufzeichnung.

---

3. Wie wird die richtige Netzwerkschnittstelle ausgewählt?

Schnittstellen ermitteln und überprüfen

Verfügbare Capture-Schnittstellen anzeigen

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

Alternativ:

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

Beispielhafte Ausgabe:

```
1. \Device\NPF_{...} (Ethernet)
2. \Device\NPF_{...} (WLAN)
3. \Device\NPF_Loopback (Adapter for loopback traffic capture)
```

Unter Linux oder macOS können Namen wie diese erscheinen:

```
1. enp0s31f6
2. wlan0
3. lo
```

oder:

```
1. en0
2. en1
3. lo0
```

Die Schnittstelle kann über ihre Nummer oder ihren Namen ausgewählt werden:

```
[TEST][PRIV] tshark -i 1
```

```
[TEST][PRIV] tshark -i en0
```

Prüfmethode

1. Die vermutete Schnittstelle auswählen.
2. Einen kurzen Mitschnitt starten.
3. Einen bekannten und autorisierten Test erzeugen, beispielsweise eine DNS-Abfrage oder einen Ping.
4. Prüfen, ob die erwarteten Pakete erscheinen.
5. Erst danach den eigentlichen Fehler reproduzieren.

Typische Fehler bei der Schnittstellenauswahl

- WLAN statt Ethernet ausgewählt;
- VPN-Schnittstelle übersehen;
- Loopback-Verkehr auf einer physischen Schnittstelle gesucht;
- virtuelle Schnittstelle einer VM oder eines Containers übersehen;

- nur den Client untersucht, obwohl der Fehler am Server oder Gateway sichtbar wird;
- auf einem Switch-Port mitgeschnitten, an dem der betreffende Verkehr nicht vorbeikommt.

> Ein leerer Mitschnitt beweist nicht, dass kein Netzwerkverkehr vorhanden war. Zuerst müssen Schnittstelle, Aufzeichnungspunkt, Berechtigung und Capture-Filter geprüft werden.

---

4. Wie wird mit Wireshark ein kontrollierter Mitschnitt erstellt?

Vorgehensweise in der grafischen Oberfläche anzeigen

Empfohlener Ablauf

1. Wireshark starten.
2. Die Schnittstelle anhand der Paketaktivität und der Systemkonfiguration auswählen.
3. Unter **Capture → Options** die Schnittstelle kontrollieren.
4. Falls erforderlich einen Capture-Filter eintragen.
5. Die Aufzeichnung starten.
6. Den Fehler möglichst einmal gezielt reproduzieren.
7. Die Aufzeichnung sofort danach beenden.
8. Die Datei im Format `pcapng` speichern.
9. Zeitpunkt, Client, Server, Testschritt und beobachtetes Verhalten dokumentieren.
10. Für die Analyse Display-Filter verwenden.

Empfohlene Dokumentation

Feld	Beispiel
Startzeit	`2026-07-31 14:02:10 CEST`
Endzeit	`2026-07-31 14:03:05 CEST`
Client	`192.0.2.10`
Server	`198.51.100.20`
Benutzer	`Max Mustermann`
Test	Anmeldung an der Beispielanwendung
Erwartung	Anmeldeseite wird geladen
Beobachtung	Verbindungsabbruch nach etwa fünf Sekunden

Feld	Beispiel
Capture-Schnittstelle	Ethernet 1
Capture-Filter	`host 198.51.100.20`

Die Aufzeichnung sollte so kurz und zielgerichtet wie möglich sein. Dadurch sinken Speicherbedarf, Analyseaufwand und Datenschutzrisiko.

---

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

Filterarten und wichtige Unterschiede anzeigen

Eigenschaft	Capture-Filter	Display-Filter
Zeitpunkt	Während der Aufzeichnung	Während oder nach der Analyse
Wirkung	Nicht passende Pakete werden nicht aufgezeichnet	Pakete bleiben in der Datei, werden aber ausgeblendet
Sprache	libpcap-/BPF-Syntax	Wireshark-Display-Filter-Syntax
TShark-Option	`-f`	`-Y`
Nachträglich änderbar	Nein	Ja
Leistungsbedarf	Vergleichsweise gering	Bei Live-Analyse möglicherweise höher
Beispiel	`tcp port 443`	`tcp.port == 443`

Capture-Filter

```
tcp port 443
```

Display-Filter

```
tcp.port == 443
```

Diese beiden Filter dürfen nicht verwechselt werden.

Ein Capture-Filter verwirft alle nicht passenden Pakete bereits bei der Aufzeichnung. Fehlende Pakete können später nicht wiederhergestellt werden.

Ein Display-Filter verändert die gespeicherte Aufzeichnung nicht. Er kann jederzeit geändert oder entfernt werden.

> Für eine Live-Aufzeichnung, die gleichzeitig mit `-w` gespeichert wird, sollte die Eingrenzung mit `-f` erfolgen. Ein mit `-Y` angegebener Display-Filter ist dafür kein Ersatz.

---

6. Welche Capture-Filter werden häufig verwendet?

Capture-Filter-Tabelle anzeigen

Aufgabe	Capture-Filter
Verkehr eines IPv4-Hosts	`host 192.0.2.10`
Nur Verkehr von einem Host	`src host 192.0.2.10`
Nur Verkehr zu einem Host	`dst host 192.0.2.10`
Verkehr eines IPv4-Netzes	`net 192.0.2.0/24`
TCP-Port 443	`tcp port 443`
UDP-Port 53	`udp port 53`
Zielport 443	`tcp dst port 443`
Quellport 443	`tcp src port 443`
DNS über TCP oder UDP	`port 53`
DHCPv4	`udp port 67 or udp port 68`
HTTP oder HTTPS	`tcp port 80 or tcp port 443`
ICMPv4	`icmp`
ICMPv6	`icmp6`
ARP	`arp`
Host und Port kombinieren	`host 192.0.2.10 and tcp port 443`
Zwei Hosts	`host 192.0.2.10 or host 198.51.100.20`
Host ausschließen	`not host 192.0.2.10`
SSH-Verkehr ausschließen	`not tcp port 22`
Broadcast- oder Multicastverkehr	`ether broadcast or ether multicast`

Beispiel: nur HTTPS-Verkehr zu einem Testserver aufzeichnen

```
[TEST][PRIV][FILE][SENS] tshark -i 1 -f "host 198.51.100.20 and tcp port 443" -a duration:60 -w https-test.pcapng
```

Wichtige Filterregeln

- Zusammengesetzte Filter in der Shell in Anführungszeichen setzen.
- Bei `and` müssen beide Bedingungen zutreffen.
- Bei `or` genügt eine der Bedingungen.
- Mit `not` wird eine Bedingung ausgeschlossen.
- Klammern verwenden, wenn die logische Reihenfolge sonst unklar ist.
- Einen engen Filter erst nach einem kurzen Kontrollmitschnitt einsetzen.

---

7. Welche Display-Filter werden häufig verwendet?

Display-Filter-Tabelle anzeigen

Aufgabe	Display-Filter
IPv4-Adresse als Quelle oder Ziel	`ip.addr == 192.0.2.10`
Nur IPv4-Quelle	`ip.src == 192.0.2.10`
Nur IPv4-Ziel	`ip.dst == 192.0.2.10`
IPv4-Netz	`ip.addr == 192.0.2.0/24`
IPv6-Adresse	`ipv6.addr == 2001:db8::10`
TCP-Port	`tcp.port == 443`
UDP-Port	`udp.port == 53`
Mehrere TCP-Ports	`tcp.port in {80, 443, 8080}`
ARP	`arp`
ICMPv4	`icmp`
ICMPv6	`icmpv6`
DNS	`dns`
Bestimmter DNS-Name	`dns.qry.name == "example.com"`
DNS-Fehlerantworten	`dns.flags.response == 1 && dns.flags.rcode != 0`
DHCPv4	`dhcp`
HTTP-Anfragen	`http.request`
TLS-Handshake	`tls.handshake`

Aufgabe	Display-Filter
TCP-SYN ohne ACK	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
TCP-RST	`tcp.flags.reset == 1`
TCP-FIN	`tcp.flags.fin == 1`
TCP-Wiederholungsübertragung	`tcp.analysis.retransmission`
TCP-Fast-Retransmission	`tcp.analysis.fast_retransmission`
Duplicate ACK	`tcp.analysis.duplicate_ack`
Vermutetes verlorenes Segment	`tcp.analysis.lost_segment`
TCP-Zero-Window	`tcp.analysis.zero_window`
Bestimmte Paketnummer	`frame.number == 100`
Pakete ab Sekunde 10	`frame.time_relative >= 10`
Zeitbereich von Sekunde 10 bis 20	`frame.time_relative >= 10 && frame.time_relative <= 20`
Große Frames	`frame.len > 1500`
Expert-Information mit Fehlerstufe	`_ws.expert.severity == error`

Filter kombinieren

```
ip.addr == 192.0.2.10 && tcp.port == 443
```

```
dns || dhcp
```

```
ip.addr == 192.0.2.10 && (tcp.analysis.retransmission || tcp.analysis.duplicate_ack)
```

Wichtiger Hinweis zu booleschen Feldern

Dieser Filter prüft nur, ob das Feld vorhanden ist:

```
tcp.flags.syn
```

Um ausschließlich gesetzte SYN-Flags zu finden, muss der Wert geprüft werden:

```
tcp.flags.syn == 1
```

8. Wie werden mit TShark kurze und kontrollierte Mitschnitte erstellt?

TShark-Aufzeichnungsbefehle anzeigen

Aufgabe	Befehl
Schnittstellen anzeigen	`[RO] tshark -D`
Live-Verkehr auf Schnittstelle 1 anzeigen	`[TEST][PRIV][SENS] tshark -i 1`
Genau 100 Pakete erfassen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -c 100 -w capture.pcapng`
60 Sekunden aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -a duration:60 -w capture.pcapng`
Auf 100.000 KiB Dateigröße begrenzen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -a filesize:100000 -w capture.pcapng`
HTTPS-Verkehr aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -f "tcp port 443" -a duration:60 -w https.pcapng`
Verkehr eines Hosts aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -f "host 192.0.2.10" -a duration:60 -w host-test.pcapng`
DNS-Verkehr aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -f "port 53" -a duration:60 -w dns-test.pcapng`

`-c 100` beendet die Aufzeichnung nach 100 erfassten Paketen.

`-a duration:60` beendet die Aufzeichnung nach 60 Sekunden.

`-a filesize:100000` beendet sie nach ungefähr 100.000 KiB.

`-w capture.pcapng` schreibt die Rohpakete in eine Mitschnittdatei. Es handelt sich nicht um eine normale Textdatei.

Kontrollierter Beispielablauf

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

```
[TEST][PRIV][FILE][SENS] tshark -i 1 -f "host 198.51.100.20 and tcp port 443" -a duration:60 -w https-test.pcapng
```

Während der 60 Sekunden wird der Fehler genau einmal reproduziert. Anschließend wird die Datei offline ausgewertet.

---

9. Wie werden vorhandene Mitschnittdateien mit TShark ausgewertet?

Offline-Analyse anzeigen

Aufgabe	Befehl
Datei einlesen	`[RO][FILE][SENS] tshark -r capture.pcapng`
Namensauflösung deaktivieren	`[RO][FILE][SENS] tshark -n -r capture.pcapng`
Nur DNS anzeigen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns"`
Nur TCP-Port 443 anzeigen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "tcp.port == 443"`
Paketdetails anzeigen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "frame.number == 100" -V`
Hexadezimaldaten anzeigen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "frame.number == 100" -x`
Nur bestimmte Protokolldetails	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns" -O dns`

Die Option `-n` deaktiviert die Namensauflösung. Dadurch werden:

- zusätzliche DNS-Anfragen während der Analyse vermieden;
- ursprüngliche IP-Adressen unverändert angezeigt;
- Verzögerungen durch Namensauflösung reduziert;
- Verwechslungen zwischen aufgezeichneten und nachträglich aufgelösten Namen vermieden.

Pakete aus einer Datei filtern und in eine neue Datei schreiben

```
[RO][FILE][SENS] tshark -r capture.pcapng -Y "ip.addr == 192.0.2.10" -w host-only.pcapng
```

Dieser Befehl liest eine vorhandene Datei, wendet einen Display-Filter an und schreibt die passenden Pakete in eine neue Mitschnittdatei.

---

10. Wie werden gezielt Felder oder CSV-Daten ausgegeben?

Feldextraktion und CSV-Ausgabe anzeigen

Mit `-T fields` wird eine tabellarische Feldausgabe erzeugt. Die gewünschten Felder werden mit `-e` angegeben.

DNS-Abfragen extrahieren

```
[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns.qry.name" -T fields -e frame.time -e ip.src -e ip.dst -e dns.qry.name
```

TCP-Verbindungen untersuchen

```
[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "tcp" -T fields -e frame.number -e frame.time_relative -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e tcp.flags
```

CSV-Ausgabe mit Kopfzeile

```
[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns" -T fields -E header=y -E separator=, -E quote=d -e frame.time -e ip.src -e ip.dst -e dns.qry.name
```

Bedeutung der Optionen

Option	Bedeutung
`-T fields`	Gibt ausschließlich ausgewählte Felder aus
`-e Feldname`	Fügt ein Feld zur Ausgabe hinzu
`-E header=y`	Schreibt die Feldnamen als Kopfzeile
`-E separator=,`	Verwendet ein Komma als Trennzeichen
`-E quote=d`	Setzt Feldwerte in doppelte Anführungszeichen

> Bei einem deutschen Tabellenkalkulationsprogramm kann das Komma als Dezimal- oder Listentrennzeichen interpretiert werden. Die Importoptionen müssen deshalb bewusst kontrolliert werden.

---

11. Welche TShark-Statistiken helfen bei der schnellen Übersicht?

Statistikbefehle anzeigen

Aufgabe	Befehl
Protokollhierarchie	`[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z io,phs`
TCP-Konversationen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z conv,tcp`

Aufgabe	Befehl
UDP-Konversationen	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z conv,udp
IPv4-Endpunkte	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z endpoints,ip
IPv6-Endpunkte	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z endpoints,ipv6
Ethernet-Endpunkte	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z endpoints,eth
Statistik pro Sekunde	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z io,stat,1
DNS-Statistik	[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z dns,tree

Bedeutung

- **Protocol Hierarchy** zeigt die Verteilung der enthaltenen Protokolle.
- **Conversations** zeigt Kommunikationsbeziehungen zwischen zwei Endpunkten.
- **Endpoints** zeigt einzelne IP- oder MAC-Adressen und deren Datenmengen.
- **I/O Statistics** zeigt Paket- und Datenmengen über die Zeit.
- **DNS Statistics** zeigt unter anderem DNS-Typen und Antwortcodes.

In Wireshark befinden sich entsprechende Ansichten überwiegend im Menü **Statistics**.

---

12. Wie werden große oder länger laufende Mitschnitte begrenzt?

Ringpuffer und Dateibegrenzung anzeigen

Ein unbegrenzter Mitschnitt kann eine Festplatte vollständig füllen. Für längere Aufzeichnungen sollte deshalb ein Ringpuffer verwendet werden.

Zwölf Dateien mit jeweils fünf Minuten Aufzeichnungsdauer

```
[TEST][PRIV][FILE][SENS] tshark -i 1 -b duration:300 -b files:12 -w capture.pcapng
```

Damit entsteht ein Ringpuffer für ungefähr eine Stunde:

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

Wenn die zwölfte Datei vollständig ist, wird die älteste Datei überschrieben.

Fünf Dateien mit jeweils ungefähr 100.000 KiB

```
[TEST][PRIV][FILE][SENS] tshark -i 1 -b filesize:100000 -b files:5 -w capture.pcapng
```

Wichtige Regeln

- `-b files:N` benötigt zusätzlich `duration`, `interval` oder `filesize`.
- Der Zielordner muss ausreichend Speicherplatz besitzen.
- Der Ringpuffer schützt nicht vor einer zu groß gewählten Gesamtgröße.
- Nach Auftreten des Fehlers muss die Aufzeichnung rechtzeitig beendet werden, bevor relevante Dateien überschrieben werden.
- Die erzeugten Dateien können sensible Inhalte enthalten.

---

13. Wie werden Mitschnittdateien geprüft und zusammengeführt?

Capinfos und Mergecap anzeigen

Dateieigenschaften anzeigen

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

Capinfos zeigt unter anderem:

- Dateityp;
- Kapselungstyp;
- Paketanzahl;
- Dateigröße;
- Aufzeichnungsdauer;
- Zeitpunkt des ersten Pakets;
- Zeitpunkt des letzten Pakets;
- durchschnittliche Paket- und Datenrate;
- vorhandene Schnittstelleninformationen.

Kompakte tabellarische Ausgabe

```
[RO][FILE][SENS] capinfos -T capture.pcapng
```

Mehrere Dateien zusammenführen

```
[RO][FILE][SENS] mergecap -w merged.pcapng part1.pcapng part2.pcapng
```

Mergecap ordnet Pakete standardmäßig anhand ihrer Zeitstempel ein. Deshalb müssen die Systemuhren der beteiligten Aufzeichnungssysteme möglichst genau synchronisiert sein.

Vor dem Zusammenführen sollten die Zeiträume geprüft werden:

```
[RO][FILE][SENS] capinfos part1.pcapng part2.pcapng
```

---

14. Wie wird eine TCP-Verbindung systematisch analysiert?

TCP-Diagnose anzeigen

Schritt 1: Verbindungsaufbau suchen

```
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

Ein regulärer TCP-Verbindungsaufbau besteht vereinfacht aus:

Schritt	Richtung	Flags
1	Client → Server	SYN
2	Server → Client	SYN, ACK
3	Client → Server	ACK

Interpretation

Beobachtung	Mögliche Ursache
SYN wird wiederholt, aber kein SYN/ACK sichtbar	Server nicht erreichbar, Firewall verwirft Pakete, Routingproblem oder falscher Aufzeichnungspunkt
SYN wird mit RST beantwortet	Zielsystem erreichbar, aber Port geschlossen oder Dienst lehnt Verbindung ab
SYN/ACK sichtbar, abschließendes ACK fehlt	Rückwegproblem, Clientproblem oder Filter-/Aufzeichnungslücke
Drei-Wege-Handshake vollständig	Grundlegender TCP-Verbindungsaufbau war erfolgreich
RST während der Sitzung	Verbindung wurde von einem Endpunkt oder einem Zwischensystem abgebrochen
FIN/ACK-Sequenz	Geordneter Verbindungsabbau

Hilfreiche Filter

```
tcp.flags.reset == 1
```

```
tcp.analysis.retransmission
```

```
tcp.analysis.duplicate_ack
```

```
tcp.analysis.zero_window
```

```
tcp.analysis.window_full
```

TCP-Stream isolieren

Nach Auswahl eines TCP-Pakets kann in Wireshark verwendet werden:

```
Analyze → Follow → TCP Stream
```

Alternativ lässt sich nach einer Streamnummer filtern:

```
tcp.stream == 0
```

> Eine TCP-Wiederholungsübertragung ist zunächst eine Beobachtung und noch keine eindeutige Ursache. Sie kann durch echten Paketverlust, verspätete Pakete, einen unvollständigen Mitschnitt, asymmetrisches Routing oder den Aufzeichnungspunkt entstehen.

---

15. Wie werden DNS-Probleme untersucht?

DNS-Diagnose anzeigen

Grundfilter

```
dns
```

Nur DNS-Anfragen

```
dns.flags.response == 0
```

Nur DNS-Antworten

```
dns.flags.response == 1
```

Bestimmten Namen untersuchen

```
dns.qry.name == "example.com"
```

Fehlerantworten anzeigen

```
dns.flags.response == 1 && dns.flags.rcode != 0
```

Typische Beobachtungen

Beobachtung	Mögliche Bedeutung
Anfrage sichtbar, aber keine Antwort	DNS-Server nicht erreichbar, Paketverlust, Firewall oder falscher Rückweg
Antwort `NXDOMAIN`	Der angefragte Name existiert aus Sicht des antwortenden DNS-Servers nicht
Antwort `SERVFAIL`	DNS-Server konnte die Anfrage nicht erfolgreich verarbeiten
Mehrere identische Anfragen	Antwort fehlt, kommt verspätet oder wird vom Client nicht akzeptiert
Antwort enthält unerwartete Adresse	Falscher DNS-Eintrag, falscher DNS-Server, Split-DNS oder Cacheproblem
Lange Zeit zwischen Anfrage und Antwort	Verzögerter DNS-Server, Weiterleitungs- oder Netzwerkproblem

TShark-Auswertung

```
[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns" -T fields -e frame.time_relative -e ip.src -e ip.dst -e dns.id -e dns.qry.name -e dns.flags.response -e dns.flags.rcode
```

Bei DNS-Problemen müssen Paketmitschnitt und lokale Resolver-Konfiguration gemeinsam betrachtet werden. Ein korrekter DNS-Austausch beweist noch nicht, dass die Anwendung das Ergebnis richtig verarbeitet hat.

---

16. Wie werden DHCP-Probleme untersucht?

DHCP-Diagnose anzeigen

Display-Filter

```
dhcp
```

Capture-Filter

```
udp port 67 or udp port 68
```

Der reguläre DHCPv4-Ablauf wird häufig als DORA bezeichnet:

Reihenfolge	Nachricht	Richtung
1	Discover	Client sucht DHCP-Server
2	Offer	DHCP-Server bietet eine Konfiguration an
3	Request	Client fordert das Angebot an
4	ACK	Server bestätigt die Zuweisung

Interpretation

Beobachtung	Mögliche Ursache
Discover, aber kein Offer	DHCP-Server nicht erreichbar, Relay fehlt, VLAN falsch oder Adresspool erschöpft
Offer sichtbar, aber kein Request	Client akzeptiert das Angebot nicht oder Mitschnitt ist unvollständig
Request, aber kein ACK	Serverproblem, Relayproblem oder Paketverlust

Beobachtung	Mögliche Ursache
DHCP NAK	Angeforderte Adresse oder Konfiguration wurde abgelehnt
Mehrere unterschiedliche Offers	Mehrere DHCP-Server antworten; möglicherweise unerwünschter DHCP-Server
DORA vollständig	DHCP-Kommunikation war grundsätzlich erfolgreich

Bei VLAN-übergreifendem DHCP muss zusätzlich der DHCP-Relay beziehungsweise IP-Helper untersucht werden.

---

17. Wie werden TLS- und HTTPS-Probleme untersucht?

TLS-Diagnose anzeigen

TLS-Verkehr

```
tls
```

TLS-Handshake

```
tls.handshake
```

TLS-Warnungen und Fehler

```
tls.alert_message
```

TCP-Port 443 zusammen mit TLS

```
tcp.port == 443 && tls
```

Typischer vereinfachter Ablauf

1. TCP-Verbindung wird aufgebaut.
2. Client sendet einen TLS-ClientHello.
3. Server sendet einen TLS-ServerHello.
4. Zertifikats- und Schlüsselparameter werden ausgetauscht.
5. Die verschlüsselte Anwendungskommunikation beginnt.

Beobachtungen

Beobachtung	Mögliche Bedeutung
TCP-Handshake fehlt	Noch kein TLS-Problem; zuerst Netzwerk und TCP untersuchen
ClientHello ohne ServerHello	Server, Firewall, TLS-Proxy oder Rückweg prüfen
TLS Alert	TLS-Seite meldet einen Protokoll- oder Zertifikatsfehler
ServerHello vorhanden, späterer Abbruch	Zertifikat, Cipher Suites, TLS-Version oder Anwendung prüfen
Verschlüsselte Application Data	TLS-Verbindung wurde mindestens bis zur verschlüsselten Datenübertragung aufgebaut

Der Inhalt moderner HTTPS-Verbindungen ist verschlüsselt. Ohne ausdrücklich autorisierte Schlüssel- oder Debug-Daten sind hauptsächlich Metadaten, TCP-Verhalten und Teile des TLS-Handshakes sichtbar.

---

18. Welche Analysefehler treten bei Wireshark besonders häufig auf?

Fehlinterpretationen und Gegenmaßnahmen anzeigen

Fehler	Warum problematisch?	Richtige Reaktion
Capture- und Display-Filter verwechselt	Erwartete Pakete werden nicht gespeichert oder Filter funktioniert nicht	Syntax und Einsatzzeitpunkt prüfen
Falsche Schnittstelle gewählt	Relevanter Verkehr fehlt	Kurzen Kontrollmitschnitt durchführen
Zu enger Capture-Filter	Entscheidende Begleitpakete fehlen	Zunächst breiter und zeitlich begrenzt aufzeichnen
Zu langer Mitschnitt	Große Dateien und hoher Analyseaufwand	Fehler gezielt reproduzieren und sofort stoppen
Namensauflösung aktiviert	Adressen erscheinen verändert oder Analyse erzeugt zusätzliche Anfragen	Für reproduzierbare CLI-Ausgaben `-n` verwenden
Jedes TCP-Retransmission-Ereignis als Paketverlust bewertet	Analyseindikatoren können durch unvollständige Mitschnitte entstehen	Sequenznummern, Aufzeichnungspunkt und Gegenmitschnitt prüfen
Rote oder schwarze Markierung als Beweis betrachtet	Farben beruhen auf konfigurierten Regeln	Paketdetails und tatsächliche Felder untersuchen

Fehler	Warum problematisch?	Richtige Reaktion
Checksum-Fehler sofort als Netzfehler bewertet	Checksum Offloading kann auf dem sendenden Host scheinbar fehlerhafte Prüfsummen erzeugen	Gegenstelle oder Netzwerkpfad vergleichen
Fehlende Pakete als Beweis für eine Firewall gewertet	Falscher Aufzeichnungspunkt oder asymmetrischer Weg möglich	Auf beiden Seiten oder an einem geeigneten Übergabepunkt mitschneiden
Promiscuous Mode überschätzt	Ein Switch sendet fremden Unicast-Verkehr nicht automatisch an jeden Port	Autorisierten Mirror-/SPAN-Port oder TAP verwenden
Zeitstempel verschiedener Systeme ungeprüft verglichen	Abweichende Systemuhren verfälschen die Reihenfolge	NTP-Status und Zeitzonen prüfen
Verschlüsselten Inhalt erwartet	TLS schützt die Nutzdaten	Handshake, Metadaten und autorisierte Logs auswerten

Checksum Offloading

Bei ausgehenden Paketen kann das Betriebssystem die endgültige Prüfsumme erst durch die Netzwerkkarte berechnen lassen. Ein Mitschnitt auf dem sendenden System kann das Paket vor dieser Berechnung erfassen. Wireshark kennzeichnet die Prüfsumme dann möglicherweise als fehlerhaft, obwohl sie auf dem Übertragungsmedium korrekt war.

TCP-Analysekennzeichnungen

Filter wie `tcp.analysis.retransmission` sind Ergebnisse der Wireshark-Analyse. Sie sind wertvolle Hinweise, aber keine automatische Ursachenfeststellung.

---

19. Wo muss der Paketmitschnitt durchgeführt werden?

Geeigneten Aufzeichnungspunkt bestimmen

Der geeignete Mitschnittpunkt hängt vom vermuteten Fehlerbereich ab.

Vermuteter Fehler	Geeigneter Mitschnittpunkt
Anwendung auf dem Client	Direkt auf dem Client
Server antwortet nicht	Client und möglichst Server
Firewall oder Routing	Vor und hinter dem betroffenen Übergang
DHCP über mehrere VLANs	Client-VLAN, Relay und Servernetz
DNS-Auflösung	Client und gegebenenfalls DNS-Server

Vermuteter Fehler	Geeigneter Mitschnittpunkt
VPN-Verbindung	Physische und virtuelle VPN-Schnittstelle
Containerkommunikation	Host-, Bridge- und gegebenenfalls Containerschnittstelle
Virtuelle Maschine	Gastbetriebssystem und gegebenenfalls virtueller Switch
Sporadischer Paketverlust	Beide Endpunkte oder geeigneter Netzwerk-TAP

In einem geswitchten Ethernet-Netz sieht ein Endgerät normalerweise:

- eigene ein- und ausgehende Pakete;
- Broadcastverkehr;
- bestimmten Multicastverkehr;
- nicht automatisch den gesamten Unicastverkehr anderer Systeme.

Für fremden Unicastverkehr ist ein autorisierter Mirror-/SPAN-Port, Netzwerk-TAP oder ein Mitschnitt direkt auf einem Endpunkt erforderlich.

Vergleichsmethode

```
Client-Mitschnitt → Netzwerkübergang → Server-Mitschnitt
```

Wenn ein Paket im Client-Mitschnitt vorhanden ist, aber im Server-Mitschnitt fehlt, liegt die Unterbrechung wahrscheinlich zwischen diesen Punkten. Für eine belastbare Aussage müssen Filter, Zeitstempel und Aufzeichnungsvollständigkeit geprüft werden.

---

20. Wie sieht ein sicherer Diagnoseablauf mit Wireshark oder TShark aus?

Empfohlene Schrittfolge anzeigen

Vorbereitung

1. Störung und erwartetes Verhalten dokumentieren.
2. Client, Server, IP-Adressen, Ports und Protokolle bestimmen.
3. Fehlerzeitpunkt und reproduzierbaren Test festlegen.
4. Berechtigung für den Mitschnitt prüfen.
5. Zeitsynchronisation der beteiligten Systeme prüfen.
6. Speicherort und Speicherplatz kontrollieren.

Aufzeichnung

7. Richtige Schnittstelle mit `tshark -D` oder Wireshark bestimmen.
8. Einen kurzen Kontrollmitschnitt durchführen.
9. Einen angemessenen Capture-Filter setzen.
10. Zeit-, Paket- oder Dateigrößenbegrenzung festlegen.
11. Aufzeichnung starten.
12. Fehler genau einmal reproduzieren.
13. Aufzeichnung sofort beenden.
14. Datei eindeutig benennen und sicher speichern.

Analyse

15. Zuerst Protokollhierarchie, Endpunkte und Konversationen prüfen.
16. Relevante Client-Server-Kommunikation isolieren.
17. ARP, DHCP und DNS vor TCP und Anwendung betrachten.
18. TCP-Verbindungsaufbau, Abbruch und Wiederholungen prüfen.
19. Anschließend TLS- oder Anwendungsprotokoll untersuchen.
20. Zeitstempel mit Server-, Firewall- und Anwendungslogs vergleichen.
21. Beobachtung und mögliche Ursache voneinander trennen.

Validierung

22. Hypothese durch einen kontrollierten Vergleichstest prüfen.
23. Falls erforderlich, Mitschnitte an zwei Punkten vergleichen.
24. Nach einer Änderung denselben Test erneut durchführen.
25. Ergebnis, Änderung und Nachweis dokumentieren.

---

21. Wie müssen Paketmitschnitte geschützt werden?

Datenschutz- und Sicherheitsregeln anzeigen

Paketmitschnitte können enthalten:

- interne und externe IP-Adressen;
- MAC-Adressen;
- DNS-Anfragen;
- Hostnamen;
- Benutzernamen;
- unverschlüsselte Passwörter;
- Cookies und Sitzungstoken;
- E-Mail- oder Chat-Inhalte;
- übertragene Dateien;
- Zertifikatsinformationen;
- Informationen über interne Systeme und Dienste.

Mindestmaßnahmen

- nur mit dokumentierter Berechtigung aufzeichnen;
- Umfang und Dauer minimieren;
- möglichst früh einen geeigneten Capture-Filter verwenden;
- Dateien ausschließlich in geschützten Verzeichnissen speichern;
- Zugriff auf zuständige Personen beschränken;
- Dateien nicht unkontrolliert per E-Mail oder Messenger versenden;
- für externe Analysen nur notwendige Pakete bereitstellen;
- Aufbewahrungs- und Löschfristen beachten;
- Weitergabe und Bearbeitung dokumentieren;
- nicht mehr benötigte Mitschnittdateien sicher löschen.

Das Öffnen von **Follow TCP Stream**, **Follow HTTP Stream** oder ähnlichen Ansichten kann zusammenhängende Anwendungsdaten sichtbar machen. Diese Funktionen dürfen deshalb nur bei entsprechender Berechtigung verwendet werden.

---

22. Kurzreferenz

Die wichtigsten Befehle und Filter anzeigen

TShark-Befehle

Aufgabe	Befehl
Version	`[RO] tshark --version`
Schnittstellen	`[RO] tshark -D`
100 Pakete aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -c 100 -w capture.pcapng`
60 Sekunden aufzeichnen	`[TEST][PRIV][FILE][SENS] tshark -i 1 -a duration:60 -w capture.pcapng`
Capture-Filter anwenden	`[TEST][PRIV][FILE][SENS] tshark -i 1 -f "tcp port 443" -a duration:60 -w capture.pcapng`
Datei lesen	`[RO][FILE][SENS] tshark -n -r capture.pcapng`
Display-Filter anwenden	`[RO][FILE][SENS] tshark -n -r capture.pcapng -Y "dns"`
Protokollhierarchie	`[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z io,phs`
TCP-Konversationen	`[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z conv,tcp`
IPv4-Endpunkte	`[RO][FILE][SENS] tshark -n -r capture.pcapng -q -z endpoints,ip`
Dateiinformationen	`[RO][FILE][SENS] capinfos capture.pcapng`
Dateien zusammenführen	`[RO][FILE][SENS] mergecap -w merged.pcapng part1.pcapng part2.pcapng`

Wichtige Capture-Filter

Aufgabe	Filter
Host	`host 192.0.2.10`
Netz	`net 192.0.2.0/24`
HTTPS	`tcp port 443`
DNS	`port 53`
DHCP	`udp port 67 or udp port 68`
ICMPv4	`icmp`
ICMPv6	`icmp6`
ARP	`arp`

Wichtige Display-Filter

Aufgabe	Filter
Host	`ip.addr == 192.0.2.10`
HTTPS-Port	`tcp.port == 443`
DNS	`dns`
DHCP	`dhcp`
TLS-Handshake	`tls.handshake`
TCP-Verbindungsbeginn	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
TCP-Abbruch	`tcp.flags.reset == 1`
Wiederholungsübertragung	`tcp.analysis.retransmission`
Duplicate ACK	`tcp.analysis.duplicate_ack`
TCP-Stream	`tcp.stream == 0`

Merksätze

- Erst Aufzeichnungspunkt und Schnittstelle prüfen, dann den Fehler mitschneiden.
- Capture-Filter bestimmen, was gespeichert wird; Display-Filter bestimmen, was angezeigt wird.
- Ein fehlendes Paket ist nur dann aussagekräftig, wenn Aufzeichnungspunkt, Filter und Mitschnitt vollständig geprüft wurden.
- Wireshark-Markierungen und TCP-Analysehinweise sind Indizien, keine fertige Fehlerursache.

- Kurze, reproduzierbare und dokumentierte Mitschnitte sind besser als stundenlange unkontrollierte Aufzeichnungen.
- Paketmitschnitte sind sensible Diagnosedaten und müssen entsprechend geschützt werden.
- Die Paketaufzeichnung zeigt, was auf dem untersuchten Netzwerkabschnitt sichtbar war – nicht automatisch den vollständigen Kommunikationsweg.

Quellen

- Wireshark User’s Guide – Capturing Live Network Data
- Wireshark User’s Guide – Capture Filters
- Wireshark User’s Guide – Display Filters
- Wireshark Display Filter Reference
- Offizielle TShark-Dokumentation
- Offizielle Dumpcap-Dokumentation
- Offizielle Capinfos-Dokumentation
- Offizielle Mergecap-Dokumentation
- Offizielle Editcap-Dokumentation
