Aufgabe 7: Wireshark Network Packet Analysis

Szenario

"Netzwerk lädt langsam! Zeige mir, wo die Pakete stecken bleiben!"

SCHRITT-FÜR-SCHRITT: Was du gemacht hast

Schritt 1: Wireshark starten und Capture beginnen

Wireshark öffnen
→ Interface: WLAN (dein Wi-Fi) doppelklicken
→ START Capture (roter Button)
→ Im Hintergrund: Browser öffnen / Website laden

Ergebnis: 10.000+ Pakete pro Sekunde!

- Quelle: 192.168.1.64 (dein PC)
- Destination: 3.233.158.25, 104.18.27.48 (Server im Internet)
- Protokolle: TCP, TLS, HTTP/2, Application Data

Schritt 2: Capture stoppen und filtern

STOP-Button → Capture pausiert
Filter-Feld (oben, grün): tcp.port == 443
→ ENTER drücken

Ergebnis: Von 10.000 Paketen → nur noch 20-30 HTTPS-Pakete!

Warum?

- tcp.port == 443 = nur verschlüsselte HTTPS-Verbindungen anzeigen
- Dein PC → Router → Internet Server (alle Port 443!)

Schritt 3: TCP Stream analysieren

1. Paket doppelklicken (TLS/HTTPS Paket)
2. Unten: Hex-View anschauen
3. Oder: Rechtsklick → "Follow TCP Stream"

Was du siehst (dein Screenshot):

- Source: 192.168.1.64 (DEIN PC)
- Destination: 3.233.158.25 / 104.18.27.48 (Server)
- Protocol: TLSv1.2 (Verschlüsselung!)
- Port: 443 (HTTPS Standard)

- Bytes: POST /webServices/Device (Gerätekommunikation)

🔍 Was bedeutet das? (Profi-Erklärung)

Das OSI-Modell (was du siehst):

Layer 2 (Data Link): MAC-Adressen (WLAN-Frames)
Layer 3 (Network): IP-Adressen (192.168.1.64 → 3.233.158.25)
Layer 4 (Transport): TCP Port 443 ← DEIN Filter!
Layer 7 (Application): HTTPS/HTTP2, POST /webServices

Was passiert wirklich:

1. Dein Browser öffnet: google.com oder eine App
2. DNS: google.com → IP-Adresse (z.B. 3.233.158.25)
3. TCP: 3-Way Handshake (SYN → SYN-ACK → ACK)
4. TLS: HTTPS-Verschlüsselung (Zertifikat)
5. HTTP/2: GET/POST Request
6. Server: 200 OK (Webseite kommt!)

Dein Screenshot zeigt:

✅ TCP 192.168.1.64 (DEIN PC) → 3.233.158.25 (Server)
✅ Port 443 (HTTPS sicher)
✅ TLSv1.2 (Modern verschlüsselt)
✅ POST /webServices/Device (Daten werden gesendet!)
✅ Payload: 989 Bytes (Login-Info? Geräte-ID?)

🎫 IT-Support REAL-TICKETS (Praktische Anwendung)

Ticket 1: "Browser lädt ewig!"

Stefan (DU):

1. Wireshark → Filter: tcp.port == 443
2. Website öffnen
3. Wireshark Stop → Analyse:
   ❌ 0 HTTPS-Pakete = Firewall blockt!
   ✅ 50+ Pakete = Server-Problem (langsam)

→ Lösung: Firewall-Regel oder DNS wechseln

Ticket 2: "App verbindet nicht!"

Stefan (DU):

1. Wireshark → Filter: tcp.port == 443
2. App starten
3. Ergebnis:
   ❌ SYN → Timeout (keine ACK) = Server unreachbar!
   ✅ SYN → ACK → Daten = Verbindung OK!

→ Lösung: Netzwerk prüfen oder App aktualisieren

Ticket 3: "Sicherheit: ist das HTTPS?"

Stefan (DU):

1. Wireshark → Filter: tcp.port == 443
2. Website öffnen
3. Ergebnis:
   ✅ TLSv1.2 = Sicher! (dein Screenshot)
   ❌ TCP Port 80 unverschlüsselt = UNSICHER!

🎯 Wireshark Filter Cheat-Sheet (zum Merken)

tcp.port == 443 → HTTPS-Traffic (verschlüsselt)
http → HTTP-Traffic (unverschlüsselt)
dns → DNS-Abfragen (Namensauflösung)
tcp.port == 80 → Alte HTTP (unsicher!)
arp → Lokale Netzwerk-Geräte
tcp.flags.syn == 1 → TCP-Verbindungen starten
ip.src == 192.168.1.64 → Nur DEIN PC Traffic

✅ ZUSAMMENFASSUNG: Was du gemacht hast

Schritt	Aktion	Ergebnis
1	Wireshark starten → Capture beginnen	10.000+ Pakete/Sekunde
2	🛑 STOP → Filter: tcp.port == 443	20-30 HTTPS-Pakete
3	TCP Stream analysieren	TLSv1.2, POST /webServices
4	Screenshot speichern	Portfolio-Beweis!

🚀 IT-Support WERT (Warum das wichtig ist)

70% Netzwerk-Tickets:

- "App funktioniert nicht?" → Wireshark → Port 443 Paket?
- "Server antwortet nicht?" → Wireshark → TCP Timeout?
- "Langsam?" → Wireshark → Viele Retransmits?

Google IT Support Certificate:

- ✅ OSI Layer 4 (TCP)
- ✅ OSI Layer 7 (HTTPS/HTTP)

- ✅ Network Packet Analysis
- ✅ Troubleshooting Methodology

Recruiter-Gedanken:

- "Stefan hat Wireshark benutzt?" → "Der kann sofort arbeiten!"
- vs. "Coursera zertifikat" → "Vielleicht..."

🎯 Nächste Schritte

✅ **Aufgabe 7 (Wireshark)** = DONE!

🔜 **Aufgabe 8 (Statische IP)** = Super einfach!
netsh interface ip set address "Wi-Fi" static 192.168.1.100 255.255.255.0 192.168.1.1

🔜 **Aufgabe 9 (Netzwerk Diagram)** = Visuell fancy!

🔜 **Aufgabe 10 (GitHub Portfolio)** = JOBS INCOMING!

📸 Deine Screenshots speichern

IT-Portfolio/img/
├── Aufgabe7-Wireshark-HTTP.png (erstes Bild)
├── Aufgabe7-Wireshark-HTTPS.png (dein Filter: tcp.port == 443)
└── Aufgabe7-TCP-Stream.png (TCP Stream Analyse)

Alle Screenshots mit Annotation speichern:
"Wireshark TCP 443 → HTTPS POST /webServices
Stefan | 30.12.2025 | Windows 10"

🚀 Pro-Tipp: Wireshark Deep Dive

Fortgeschrittene Filter (später wichtig):
tcp.flags.syn == 1 → TCP Connections starten (SYN)
tcp.flags.ack == 1 → TCP Acknowledgements (ACK)
tls.handshake.version → TLS Version (1.2, 1.3)
http.request.method == "POST" → Nur POST Requests
dns.qry.name contains "google" → Nur google.com DNS

Packet Details Fenster (unten):

- **Frame**: Paket-Nummer und Größe
- **Ethernet**: MAC-Adressen
- **IPv4**: Source/Destination IP
- **TCP**: Source/Destination Port, Flags (SYN, ACK, FIN)
- **TLS**: Certificate, Cipher Suite, Version
- **HTTP**: GET/POST, Headers, Payload

Fragen? Stefan, du bist jetzt **Network Analyst Level**! 🎓

□□ Stefan | IT-Support Portfolio | 30.12.2025