© IT-Läraren (itlararen.se)

Laboration- Undersök OSI-modellen och TCP/IP-lagermodellen med Wireshark

Syfte: Att med hjälp av Wireshark koppla verkliga paket/frames till OSI-modellens lager (1–7) och till TCP/IP-modellen (Länk, Internet, Transport, Applikation).

Lärandemål (efter labben kan du):

- Identifiera vilka protokoll/fält som hör till lager 2–4 (Ethernet/ARP/IP/TCP/UDP) och lager 7 (DNS/HTTP/TLS) i en inspelning.
- Använda grundläggande display filters och följa en konversation (t.ex. TCP-ström).
- Förklara inkapsling: hur data får nya headers per lager och hur det syns i Wireshark.
- Koppla OSI-lagren till TCP/IP-modellen i en sammanhängande kedja för en webbförfrågan.

Förutsättningar och viktiga regler

- **Förkunskaper:** Du känner igen grundbegrepp som MAC/IP, portnummer, TCP vs UDP, DNS, HTTP/HTTPS.
- **Integritet:** Spela bara in din egen trafik i en kontrollerad miljö. Undvik att logga in på tjänster med riktiga lösenord under inspelning.
- **Tid:** 45–90 minuter.
- **Verktyg:** Wireshark är installerat och du kan starta en live capture på rätt nätverksinterface.

Förberedelser - starta och spara en inspelning

1. Öppna Wireshark och välj det interface som har tydlig trafik (Wi-Fi/Ethernet med rörlig aktivitetsgraf).
2. Klicka *Start* och spela in i 1–3 minuter medan du genomför uppgifterna nedan.
3. Stoppa inspelningen och spara filen som **osi_lab.pcapng**.
4. Tips: Om du får väldigt mycket bakgrundstrafik, fortsätt ändå—du filtrerar fram relevant trafik i nästa steg.

Uppgift 1 – Lager 2 (Datalänk): Ethernet och ARP

Mål: Se hur en Ethernet-ram ser ut och hur ARP används för att mappa IP → MAC på ett lokalt nät.

1. **Generera ARP-trafik:** Öppna en terminal och kör **ping** mot din default gateway (t.ex. 192.168.0.1). Om du redan har en ARP-cache kan du behöva vänta en stund eller pinga en annan lokal adress i samma subnet.
2. Skriv display filter: **arp**
3. Välj ett **ARP request**-paket och expandera delarna i paketdetaljerna: **Ethernet II** och **Address Resolution Protocol**.
4. Svara i din rapport:
    - Vilken är **Destination MAC** i ARP request (och varför)?
    - Vilken är **Sender MAC** och **Sender IP**?
    - Hittar du ett matchande **ARP reply**? Vilka fält ändras jämfört med request?

5. Koppla till OSI: Motivera kort varför Ethernet/ARP hör hemma i **lager 2** (och vilka delar av OSI som Wireshark inte kan "se" direkt i en vanlig capture).

Uppgift 2 – Lager 3 (Nätverk): IPv4/IPv6 och ICMP

Mål: Se IP-headern och hur ICMP används vid ping (Echo request/reply) samt förstå TTL/hop-konceptet.

1. Kör **ping** mot en extern adress/ett domännamn (t.ex. **ping 1.1.1.1** eller en webbplats ni får använda).
2. Använd display filter: **icmp** (för IPv4) eller **icmpv6** (för IPv6).
3. Välj en **Echo (ping) request** och expandera **Internet Protocol Version 4** (eller IPv6).
4. Svara i din rapport:
    - Vilken är **Source IP** och **Destination IP**?
    - Vilket värde har **Time to live (TTL)** i request? Hur tolkar du det?
    - Vilken skillnad ser du mellan request och reply (t.ex. ICMP-typ, checksumma, identifier/sekvens)?
5. Koppla till OSI: Förklara kort varför IP/ICMP är **lager 3** (routing, logiska adresser).

Uppgift 3 – Lager 4 (Transport): TCP 3-way handshake och portar

Mål: Identifiera TCP-anslutningens uppbyggnad (SYN, SYN/ACK, ACK) och koppla portar till applikationer.

1. Öppna en webbsida i webbläsaren (gärna en enkel sida) medan inspelning pågår.
2. Använd display filter för att hitta handshakes: **tcp.flags.syn == 1**
3. Välj en SYN-ram och notera i **Transmission Control Protocol**: **Source Port**, **Destination Port**, **Seq** (sekvensnummer) och flaggor.
4. Hitta de två följande paketen i samma flöde (SYN/ACK och ACK). Tips: högerklicka paketet → *Conversation Filter* → *TCP*, eller filtrera med **tcp.stream == X** (där X är strömmens nummer).
5. Svara i din rapport:
    - Vilken port användes som destination? Vad säger den porten om applikationsprotokollet?
    - Hur ser flaggorna ut i SYN, SYN/ACK och ACK?
    - Varför behövs 3-way handshake ur tillförlitlighetssynpunkt?
6. Koppla till OSI: Förklara varför TCP hör hemma i **lager 4** (änd-till-änd, portar, flödeskontroll).

Uppgift 4 – Lager 7 (Applikation): DNS och HTTP/HTTPS (TLS)

Mål: Följa en typisk "gå till en webbplats"-kedja: DNS-uppslag → TCP-anslutning → (ev.) TLS-handshake → HTTP-förfrågan/svar.

A) DNS – hitta vilken IP ett namn blir

1. Besök en webbplats du inte har öppnat nyligen (för att öka chansen att DNS sker under capture).
2. Filtrera: **dns**
3. Välj en **Standard query** och expandera **Domain Name System**.
4. Svara i din rapport:

- Vilket är **Query Name** (domännamnet)?
- Är transporten **UDP** eller **TCP**? Vilken port används?
- Hitta motsvarande **Standard query response**: vilken/vilka **A**- eller **AAAA**-poster returneras?

B) HTTP eller HTTPS – vad kan du se i klartext?

Välj ett av alternativen nedan (A rekommenderas om ni har en kontrollerad testserver):

1. **Alternativ 1: HTTP (okrypterat)**
   - Öppna en sida som kör HTTP (utan inloggning och utan känslig info).
   - Filtrera: **http** (eller **tcp.port == 80**).
   - Välj en **GET**-request och expandera **Hypertext Transfer Protocol**.
   - Rapport: notera **Host**, **Request URI**, **User-Agent** och serverns **Status Code** i svaret.
2. **Alternativ 2: HTTPS (TLS)**
   - Öppna en vanlig HTTPS-webbplats.
   - Filtrera: **tls** (eller **tcp.port == 443**).
   - Hitta en **Client Hello** (filter: **tls.handshake.type == 1**) och expandera **Transport Layer Security**.
   - Rapport: notera vilka delar som fortfarande kan synas utan dekryptering (t.ex. servernamn/SNI om det finns, TLS-version, cipher suites) och vilka delar som inte syns (HTTP-innehållet).

Uppgift 5 – Knyt ihop: OSI vs TCP/IP (lagermodeller)

Instruktion: Välj en specifik "webb-besök"-sekvens i din inspelning (DNS + anslutning till webbserver) och beskriv inkapslingen steg för steg.

- **OSI lager 7–5 (Applikation/Presentation/Session)** ↔ **TCP/IP Applikation** (t.ex. DNS, HTTP, TLS)
- **OSI lager 4 (Transport)** ↔ **TCP/IP Transport** (TCP/UDP)
- **OSI lager 3 (Nätverk)** ↔ **TCP/IP Internet** (IP, ICMP)
- **OSI lager 2–1 (Datalänk + Fysiskt)** ↔ **TCP/IP Länk** (Ethernet/Wi-Fi + fysisk överföring)

1. Skriv en kort kedja (5–10 meningar) där du beskriver vad som händer när du skriver in ett domännamn i webbläsaren, och referera till **konkreta paket** i din capture (t.ex. "i paket X ser jag DNS-frågan …").
2. För varje lager du nämner: ange minst **två fält** från Wireshark som stödjer att du är på rätt lager (ex: MAC-adresser, IP-adresser, portar, flags, HTTP-headers).
3. Reflektera: Vilka OSI-lager är svåra att "se" tydligt i Wireshark och varför?

Reflektionsfrågor (kort)

- Hur påverkar kryptering (TLS) vad du kan analysera på applikationsnivå?
- Vilka felkällor finns när man tolkar en capture (t.ex. cache, bakgrundstrafik, NAT, QUIC/HTTP3)?
- Vilka display filters tyckte du var mest användbara och varför?

Inlämning (förslag på struktur)

1. **Miljö**: OS, nätverk (Wi-Fi/Ethernet), ungefärlig tidsperiod för capture.
2. **Uppgift 1–4**: Svar på frågorna + skärmbilder/utklipp av relevanta fält (om läraren kräver det).
3. **Uppgift 5**: Din inkapslingsbeskrivning + OSI↔TCP/IP-koppling.
4. **Reflektion**: korta svar.
5. **Bilaga**: bifoga **osi_lab.pcapng** om det efterfrågas.