S11 – L2

Titolo:

Uso di Wireshark e tcpdump per osservare l’Handshake TCP a 3 vie (Mininet)

Introduzione:

In questo esercizio avvio una **topologia Mininet, genero traffico HTTP tra un client e un web server**, catturo i pacchetti con **tcpdump**, poi analizzo il **Three-Way Handshake TCP** con **Wireshark** e infine visualizzo lo stesso handshake da terminale con **tcpdump -r**.

Parte 1 — Preparare gli host per catturare il traffico

1. **Avvio la VM**

- Avvio **CyberOps Workstation**
- Accedo con:
    - user: `analyst`
    - password: `cyberops`

2. **Avvio Mininet**

- Apro un terminale e digito:

```
sudo lab.support.files/scripts/cyberops_topo.py
```

3. **Apro i terminali degli host H1 e H4**

- Nel prompt di Mininet digito:

```
xterm H1
xterm H4
```

4. **Avvio il web server su H4**

- Nella finestra di H4 (root), eseguo:

```
/home/analyst/lab.support.files/scripts/reg_server_start.sh
```

5. **Su H1 passo all’utente analyst (Firefox non parte da root)**

- Nella finestra di H1 digito:

```
su analyst
```

6. **Avvio Firefox su H1**

- Sempre su H1 (come analyst) digito:

```
firefox &
```

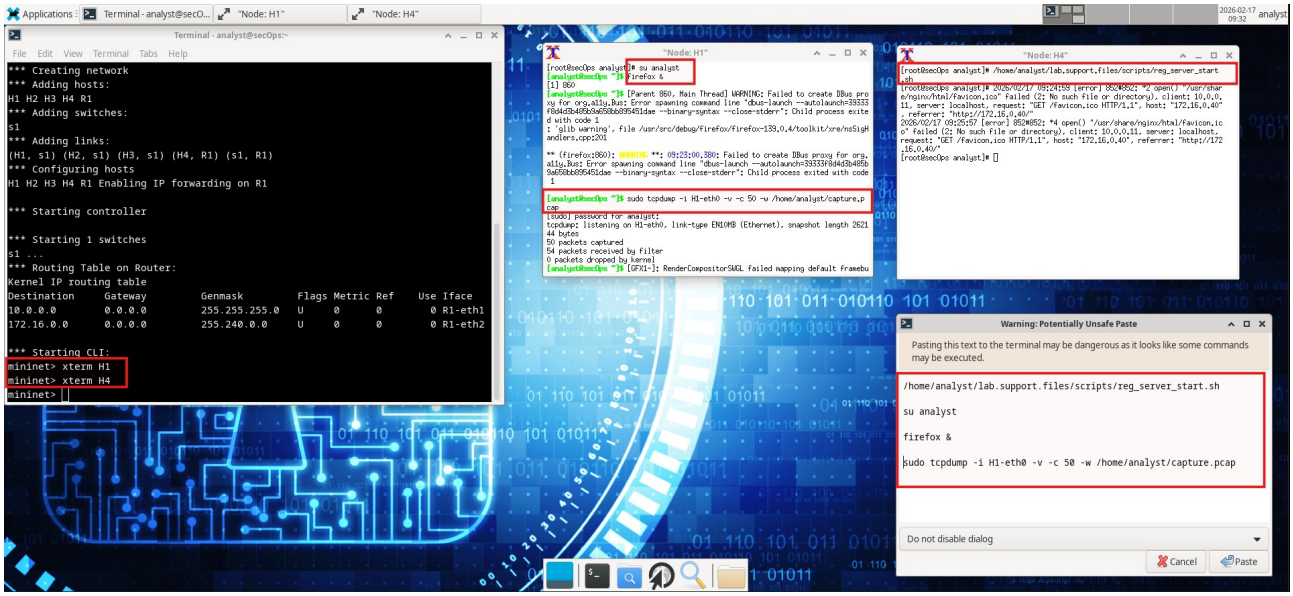
7. **Avvio la cattura con tcpdump e salvo su file**

- Nel terminale Node: H1 lancio la cattura (50 pacchetti) verso file:

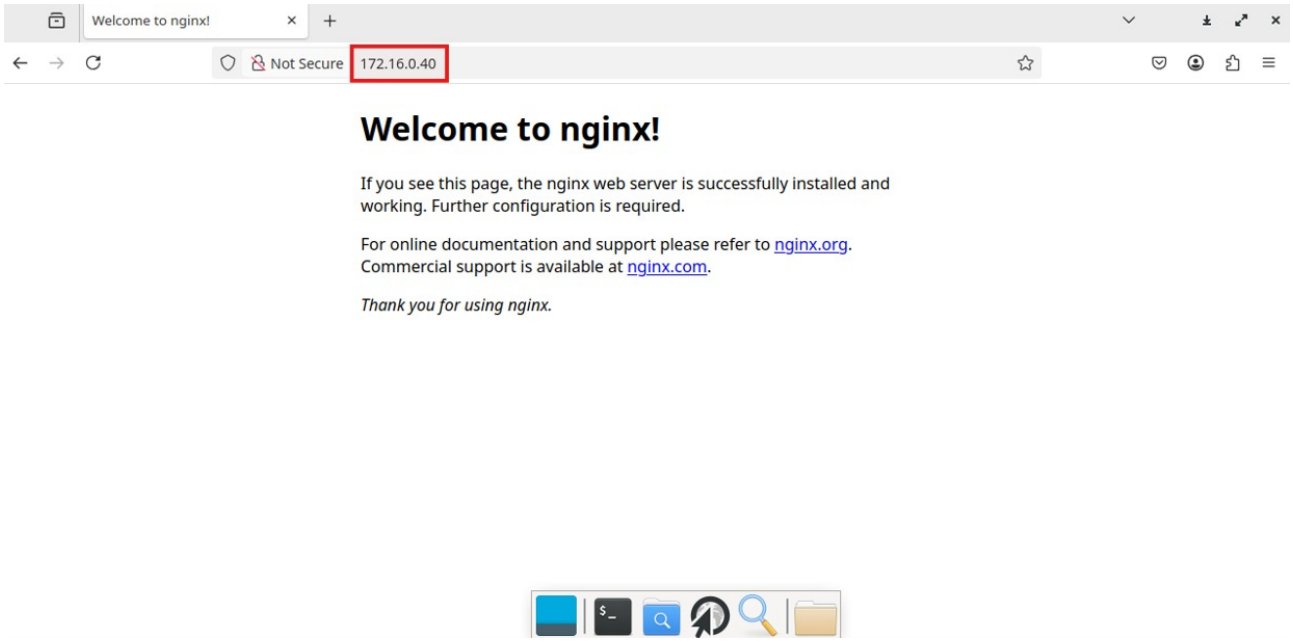
```
sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
```

8. **Genero traffico HTTP verso il server**

- Appena tcpdump è in esecuzione, nel browser vado velocemente su:
    - **`http://172.16.0.40`**



Output: 50 pacchetti catturati correttamente, traffico HTTP generato correttamente verso 172.16.0.40 e handshake TCP avvenuto con successo.



Output: “Welcome to nginx!” visualizzata correttamente su http://172.16.0.40.

Parte 2 — Analizzare i pacchetti con Wireshark

Passo 1 — Apro la cattura e filtro TCP

1. Torno al prompt e avvio Wireshark su H1:

```
wireshark-gtk &
```

- Se compare l’avviso sull’esecuzione come superutente, clicco **OK**.

2. In Wireshark:

- **File > Open**
- Seleziono: **`/home/analyst/capture.pcap`**

3. Nel filtro in alto digito:

- **`tcp`**
- Applico il filtro.

Passo 2 — Esamino i 3 pacchetti del Three-Way Handshake

> Nota pratica: se non vedo bene i dettagli, ridimensiono i riquadri di Wireshark (lista pacchetti / dettagli).

Frame 1 (inizio handshake: client → server)

- Seleziono il **primo pacchetto** del flusso.
- Espando **Transmission Control Protocol**
- Espando **Flags** e individuo il flag impostato.

Domande e Risposte:

Qual è il numero di porta TCP di origine?

- **Porta TCP sorgente:** `58716`

Come viene classificata questa porta?

- **Classificazione porta sorgente:** *effimera/dinamica (client)*

Qual è il numero di porta TCP di destinazione?

- **Porta TCP destinazione:** `80`

Come viene classificata questa porta?

- **Classificazione porta destinazione:** *well-known (HTTP)*

Quale flag TCP è impostato nel primo pacchetto?

- **Flag impostato:** `SYN`

Qual è il numero di sequenza relativo?

- **Sequence number relativo:** `0`

Questo pacchetto è il “via” dell’handshake TCP e usa il flag **SYN** per avviare la sessione.

Frame 2 (risposta server: server → client)

- Seleziono il **secondo pacchetto** (risposta del server).

Domande e Risposte:

Quali sono le porte TCP sorgente e destinazione nel pacchetto di risposta del server?

- **Porte (source/dest):** `80 → 58716`

Quali flag TCP sono impostati nel secondo pacchetto?

- **Flag impostati:** `SYN, ACK`

Qual è il numero di sequenza relativo?

- **Sequence relativo:** `0`

Qual è il numero di acknowledgment relativo?

- **Acknowledgment relativo:** `1`

Il server conferma (ACK) e sincronizza (SYN) la sessione, completando il secondo step del three-way handshake.

Frame 3 (chiusura handshake: client → server)

- Seleziono il **terzo pacchetto** (ultimo dell'handshake).

Domande e Risposte:

Quale flag TCP è impostato nel terzo pacchetto?

- **Flag impostato:** `ACK`

Quali sono i numeri relativi di sequenza e acknowledgment nel terzo pacchetto?

- I **numeri relativi** di **sequence** e **acknowledgment** risultano impostati a `1` come punto di partenza e la connessione è stabilita.

Dopo questi 3 scambi, TCP ha verificato host/servizio e sincronizzato la sessione, e può iniziare il trasferimento dati.

Parte 3 — Visualizzare i pacchetti usando tcpdump

1. **Consulto la man page**

- Apro un nuovo terminale e digito:

```
man tcpdump
```

Domanda: “Cosa fa l’opzione -r?”

- `-r` **legge i pacchetti da un file pcap** (replay/lettura offline) invece di catturarli da un’interfaccia live.

2. **Leggo dal file e mostro i primi pacchetti TCP**

- Nel terminale digito:

```
tcpdump -r /home/analyst/capture.pcap tcp -c 3
```

Se voglio vedere più righe dell’handshake, aumento **`-c`** (es. **`-c 10`**).

3. **Chiudo Mininet e pulisco**

- Torno al terminale di Mininet e digito:

```
quit
```

- Poi pulisco i processi Mininet:

```
sudo mn -c
```

Domande di Riflessione:

1. Ci sono centinaia di filtri disponibili in Wireshark. Elenca tre filtri che potrebbero essere utili a un amministratore di rete.

Risposta:

- `tcp` → per analizzare connessioni, handshake e ritrasmissioni.
- `dns` → per verificare richieste di risoluzione nomi e problemi DNS.
- `icmp` → per diagnosticare problemi di connettività (ping, unreachable, ecc.).

2. In quali altri modi Wireshark potrebbe essere utilizzato in una rete di produzione?

Risposta:

- Troubleshooting di problemi di rete (latenza, packet loss, ritrasmissioni).
- Analisi di handshake TCP incompleti o connessioni interrotte.
- Identificazione di traffico anomalo o sospetto.
- Verifica del corretto funzionamento dei protocolli applicativi (HTTP, DNS, TLS).
- Supporto ad attività di incident response e analisi forense di rete.

Conclusioni:

Ho completato il laboratorio catturando traffico reale in Mininet con **tcpdump**, analizzando il **three-way handshake** con **Wireshark** (porte, flag, sequence/ack) e verificando gli stessi pacchetti anche da terminale con `tcpdump -r`.