

WIRESHARK

L'HANDSHAKE TCP

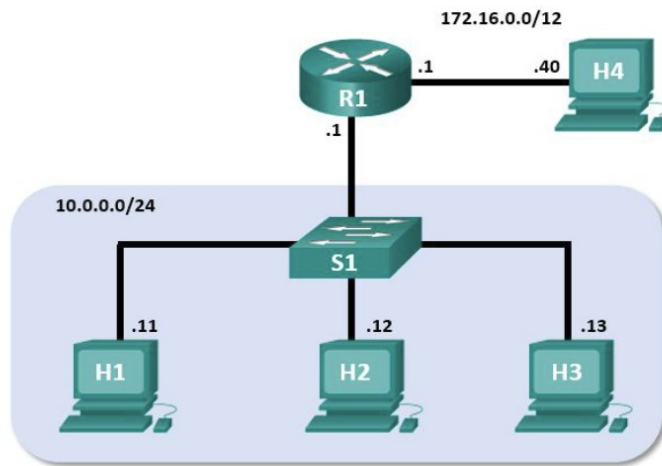
Usare Wireshark per Osservare l'Handshake a 3 Vie TCP

Obiettivi:

- **Parte 1: Preparare gli Host per Catturare il Traffico**
- **Parte 2: Analizzare i Pacchetti usando Wireshark**
- **Parte 3: Visualizzare i Pacchetti usando tcpdump**

SCENARIO

In questo laboratorio, si userà Wireshark per catturare ed esaminare i pacchetti generati tra il browser del PC che utilizza il protocollo HTTP (HyperText Transfer Protocol) e un server web, come www.google.com. Quando un'applicazione, come HTTP o FTP (File Transfer Protocol), si avvia per la prima volta su un host, TCP utilizza l'handshake a tre vie per stabilire una sessione TCP affidabile tra i due host. Ad esempio, quando un PC utilizza un browser web per navigare in internet, viene avviato un handshake a tre vie e viene stabilita una sessione tra l'host del PC e il server web. Un PC può avere più sessioni TCP attive simultaneamente con vari siti web.



Risorse Richieste: Macchina virtuale CyberOps Workstation

PARTE 1: PREPARARE GLI HOST PER CATTURARE IL TRAFFICO

ISTRUZIONI

Parte 1: Preparare gli Host per Catturare il Traffico

- Avviare la VM CyberOps. Accedere con nome utente analyst e password cyberops.
- Avviare Mininet.

```
[analyst@sec0ps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
```

- Avviare gli host H1 e H4 in Mininet.

```
*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
```

- Avviare il server web su H4.

```
[root@sec0ps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start.sh
```

Risultato

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
[sudo] password for analyst:

CyberOPS Topology:

      -----
      | R1 |-----| H4 |
      -----
      |
      |
      -----
|-----| S1 |-----|
|         |         |
|         |         |
|         |         | | | |
|---|---|---|---|---|
| H1 |   | H2 |   | H3 |
|-----|-----|

*** Adding internal links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
(H1, s1) (H2, s1) (H3, s1) (H4, R1) (s1, R1)
*** Configuring hosts
H1 H2 H3 H4 R1 Enabling IP forwarding on R1

*** Starting controller

*** Starting 1 switches
s1 ...
*** Routing Table on Router:
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U        0      0          0 R1-eth1
172.16.0.0      0.0.0.0         255.240.0.0     U        0      0          0 R1-eth2
```

Passaggi:

e. Per motivi di sicurezza, non è possibile eseguire Firefox dall'account utente root. Sull'host H1, usare il comando `su` (switch user) per passare dall'utente root all'account utente analyst:

```
[root@secOps analyst]# su analyst
```

f. Avviare il browser web su H1. Ci vorrà qualche momento.

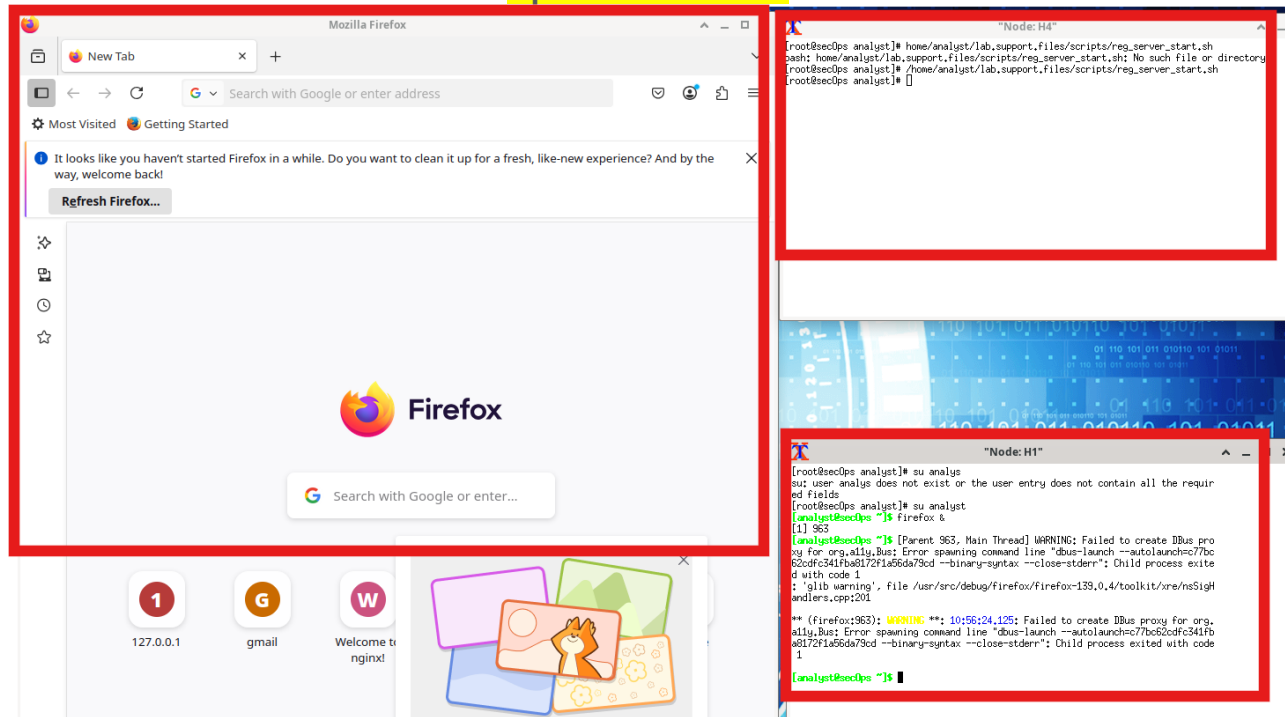
```
[analyst@secOps ~]$ firefox &
```

g. Dopo l'apertura della finestra di Firefox, avviare una sessione tcpdump nel terminale Node: H1 e inviare l'output a un file chiamato capture.pcap. Con l'opzione `-v`, è possibile osservare l'avanzamento. Questa cattura si fermerà dopo aver catturato 50 pacchetti, poiché è configurata con l'opzione `-c 50`.

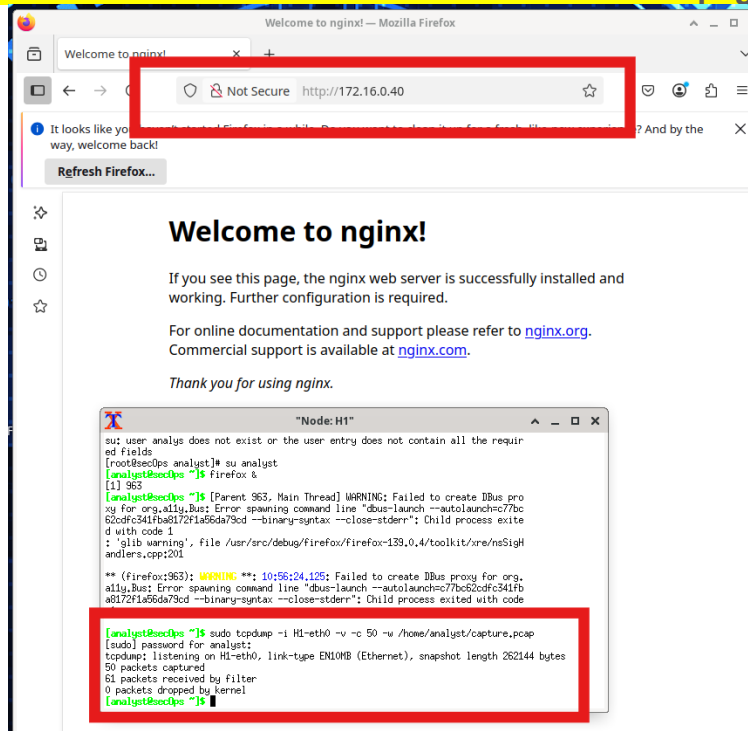
```
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
```

h. Dopo l'avvio di tcpdump, navigare rapidamente a 172.16.0.40 nel browser web Firefox.

Apertura Browser:



Cattura del traffico durante il caricamento della pagina



PARTE 2: ESAMINARE LE INFORMAZIONI ALL'INTERNO DEI PACCHETTI

Analizzare i Pacchetti usando Wireshark

Applicare un filtro alla cattura salvata.

a. Premere INVIO per vedere il prompt. Avviare Wireshark su Node: H1. Fare clic su OK quando viene richiesto l'avviso riguardante l'esecuzione di Wireshark come superutente.

[analyst@secOps ~]\$ wireshark-gtk &

b. In Wireshark, fare clic su File > Open. Selezionare il file pcap salvato situato in /home/analyst/capture.pcap.

c. Applicare un filtro tcp alla cattura. In questo esempio, i primi 3 frame rappresentano il traffico di interesse

Attivazione Wireshark:

The screenshot shows the Wireshark interface with a packet capture list on the left and a packet details pane on the right. The packet list shows several DNS and HTTP packets. The packet details pane shows the selected packet's structure, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (query). The terminal output on the right shows the command 'wireshark-gtk &' being executed, and the resulting output, including the path to the capture file and the command to run Wireshark as a background process.

Risultati Filtro:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.882541	10.0.0.11	172.16.0.40	TCP	74	39508 -> 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=343706531 TSecr=0 WS=512
8	0.882719	172.16.0.40	10.0.0.11	TCP	74	80 -> 39508 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=1028308459 TSecr=343706531 WS=512
9	0.882729	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=343706531 TSecr=1028308459
10	0.882817	10.0.0.11	172.16.0.40	HTTP	397	GET / HTTP/1.1
11	0.882869	172.16.0.40	10.0.0.11	TCP	66	80 -> 39508 [ACK] Seq=1 Ack=332 Win=43520 Len=0 TSval=1028308459 TSecr=343706531
12	0.883151	172.16.0.40	10.0.0.11	TCP	304	80 -> 39508 [PSH, ACK] Seq=1 Ack=332 Win=43520 Len=238 TSval=1028308460 TSecr=343706531 [TCP PDU reassembled in 13]
13	0.883211	172.16.0.40	10.0.0.11	HTTP	681	HTTP/1.1 200 OK (text/html)
14	0.883223	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=332 Ack=854 Win=41984 Len=0 TSval=343706532 TSecr=1028308460
15	0.972511	10.0.0.11	172.16.0.40	HTTP	411	GET /favicon.ico HTTP/1.1
16	0.972599	10.0.0.11	172.16.0.40	TCP	374	HTTP/1.1 404 Not Found (text/html)
17	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
18	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
19	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
20	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
21	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
22	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
23	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
24	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554
25	0.972599	10.0.0.11	172.16.0.40	TCP	66	39508 -> 80 [ACK] Seq=677 Ack=1162 Win=41984 Len=0 TSval=343706626 TSecr=1028308554

Passaggi:

- In questo esempio, il frame 1 è l'inizio dell'handshake a tre vie tra il PC e il server su H4. Nel riquadro dell'elenco dei pacchetti (sezione superiore della finestra principale), selezionare il primo pacchetto, se necessario.
- Fare clic sulla freccia a sinistra del Transmission Control Protocol nel riquadro dei dettagli del pacchetto per espanderlo ed esaminare le informazioni TCP. Localizzare le informazioni sulla porta di origine e destinazione.
- Fare clic sulla freccia a sinistra dei Flags. Un valore di 1 significa che il flag è impostato. Localizzare il flag impostato in questo pacchetto.

Risultati:

tcp					
No.	Time	Source	Destination	Protocol	Length Info
Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)					
Ethernet II, Src: 3a:7f:df:b9:07:67 (3a:7f:df:b9:07:67), Dst: 36:cd:ad:be:0f:4a (36:cd:ad:be:0f:4a)					
Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40					
Transmission Control Protocol, Src Port: 39508, Dst Port: 80, Seq: 0, Len: 0					
Source Port: 39508					
Destination Port: 80					
[Stream index: 0]					
[Stream Packet Number: 1]					
[Conversation completeness: Incomplete, DATA (15)]					
..0. = RST: Absent					
...0 = FIN: Absent					
.... 1... = Data: Present					
.... .1.. = ACK: Present					
.... ..1. = SYN-ACK: Present					
.... ...1 = SYN: Present					
[Completeness Flags: ..DASS]					
[TCP Segment Len: 0]					
Sequence Number: 0 (relative sequence number)					
Sequence Number (raw): 1177623379					
[Next Sequence Number: 1 (relative sequence number)]					
Acknowledgment Number: 0					
Acknowledgment number (raw): 0					
1010 = Header Length: 40 bytes (10)					
Flags: 0x002 (SYN)					
000. = Reserved: Not set					
...0 = Accurate ECN: Not set					
.... 0... = Congestion Window Reduced: Not set					
.... .0.. = ECN-Echo: Not set					
.... ..0. = Urgent: Not set					
.... ...0 = Acknowledgment: Not set					
....0 = Push: Not set					
....0 = Reset: Not set					
....1. = Syn: Set					
[Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]					
[Connection establish request (SYN): server port 80]					
[Severity level: Chat]					
[Group: Sequence]					
....0 = Fin: Not set					
[TCP Flags:S.]					
Window: 42340					
[Calculated window size: 42340]					

1. Qual è il numero di porta TCP di origine?

La porta di origine è **39508**.

2. Come classificheresti la porta di origine?

In ambito networking, le porte si classificano in base al loro numero. La porta **39508** è una **Porta Effimera (o Dinamica)**

Perché: Le porte nel range da **49152 a 65535** (standard IANA) o spesso da **1024 a 65535** (in molti sistemi operativi) sono assegnate temporaneamente dal sistema operativo client per una specifica sessione di comunicazione. Una volta chiusa la connessione, la porta torna disponibile.

3. Qual è il numero di porta TCP di destinazione?

La porta di destinazione è **80**.

4. Come classificheresti la porta di destinazione?

La porta **80** è una **Porta Ben Nota (Well-Known Port)**.

Perché: Le porte da **0 a 1023** sono riservate a servizi di sistema o protocolli standard. In questo caso, la porta 80 è universalmente assegnata al protocollo **HTTP** (traffico web non crittografato).

5. Quale flag è impostato?

Il flag impostato è il **SYN (Synchronize)**.

A cosa serve: In una analisi Blue Teaming o Red Teaming, questo pacchetto è il "primo passo" del **Three-Way Handshake** (il processo di instaurazione di una connessione TCP).

Funzione: Il client invia il flag SYN per indicare al server che vuole iniziare una comunicazione e per sincronizzare i **Sequence Numbers** (numeri di sequenza) iniziali. Senza questo scambio, i due host non saprebbero come ordinare i pacchetti di dati che si scambieranno in seguito.

6. A quale valore è impostato il numero di sequenza relativo?

Il numero di sequenza relativo è **0**.

Nota tecnica: Wireshark mostra "0" per facilitare la lettura agli analisti. Il numero reale (**Raw Sequence Number**) è **1177623379**. Wireshark sottrae questo valore base da tutti i pacchetti successivi della stessa sessione per permetterti di vedere facilmente l'incremento dei dati inviati (es. 0, 1, 150, ecc.).

d. Selezionare il pacchetto successivo nell'handshake a tre vie. In questo esempio, è il frame 2. Questa è la risposta del server web alla richiesta iniziale di avviare una sessione.

tcp					
No.	Time	Source	Destination	Protocol	Length Info
7	0.882541	10.0.0.11	172.16.0.40	TCP	74 39508 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM=1
8	0.882719	172.16.0.40	10.0.0.11	TCP	74 80 → 39508 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460
9	0.882729	10.0.0.11	172.16.0.40	TCP	66 39508 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=34370653
10	0.882817	10.0.0.11	172.16.0.40	HTTP	397 GET / HTTP/1.1

Destination Port: 39508
[Stream index: 0]
[Stream Packet Number: 2]
[Conversation completeness: Incomplete, DATA (15)]

...0... = RST: Absent
...0... = FIN: Absent
...1... = Data: Present
...1... = ACK: Present
...1... = SYN-ACK: Present
...1... = SYN: Present
[Completeness Flags: ..DASS]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 2047480671
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1177623380
1010... = Header Length: 40 bytes (10)
Flags: 0x012 (SYN, ACK)

000... = Reserved: Not set
...0... = Accurate ECN: Not set
...0... = Congestion Window Reduced: Not set
...0... = ECN-Echo: Not set
...0... = Urgent: Not set
...1... = Acknowledgment: Set
...0... = Push: Not set
...0... = Reset: Not set
...1... = Syn: Set

[Expert Info (Chat/Sequence): Connection establish acknowledge (SYN+ACK): server port 80]
[Connection establish acknowledge (SYN+ACK): server port 80]
[Severity level: Chat]
[Group: Sequence]
...0... = Fin: Not set
[TCP Flags:A..S.]
Window: 43440

7. Quali sono i valori delle porte di origine e destinazione?

Porta di origine (Src Port): 80.

Porta di destinazione (Dst Port): 39508.

In questa fase, i ruoli sono invertiti rispetto al primo pacchetto: il server (porta 80) sta rispondendo al client (porta 39508).

8. Quali flag sono impostati?

I flag impostati in questo frame sono due:

Acknowledgment (ACK): Impostato (Set). Indica che il server ha ricevuto la richiesta iniziale e sta confermando il numero di sequenza del client.

Syn (SYN): Impostato (Set). Indica che anche il server vuole sincronizzare i propri numeri di sequenza per stabilire la connessione bidirezionale.

Questo pacchetto è tecnicamente classificato come **SYN, ACK**.

9. A quali valori sono impostati i numeri relativi di sequenza e acknowledgment?

Sequence Number (relative): 0.

Acknowledgment Number (relative): 1.

e. Infine, selezionare il terzo pacchetto nell'handshake a tre vie. Esaminare il terzo e ultimo pacchetto dell'handshake.

Quale flag è impostato?

In questo pacchetto è impostato esclusivamente il flag **Acknowledgment (ACK)**.

- Come puoi vedere nel dettaglio dei flag, il bit relativo a **Acknowledgment** è impostato a **Set**, mentre il flag **Syn** è ora **Not set**.
- Questo pacchetto serve al client per confermare al server di aver ricevuto il pacchetto SYN-ACK precedente e che la connessione è ora ufficialmente stabilita.

Riepilogo dell'Handshake completo (Three-Way Handshake)

Basandoci sui frame che abbiamo analizzato:

1. **SYN (Frame 7):** Il client (10.0.0.11) chiede la connessione.
2. **SYN-ACK (Frame 8):** Il server (172.16.0.40) risponde e accetta.
3. **ACK (Frame 9):** Il client conferma la ricezione e stabilisce la sessione.

Da questo momento in poi, come si vede nel **Frame 10**, può iniziare lo scambio di dati effettivo tramite il protocollo HTTP (richiesta GET).

PARTE 3: VISUALIZZARE I PACCHETTI USANDO TCPDUMP

È anche possibile visualizzare il file pcap e filtrare per le informazioni desiderate. a. Aprire una nuova finestra di terminale, inserire `man tcpdump`. Nota: Potrebbe essere necessario premere INVIO per vedere il prompt. Utilizzando le pagine manuale (man pages) disponibili con il sistema operativo Linux, è possibile leggere o cercare tra le pagine manuale le opzioni per selezionare le informazioni desiderate dal file pcap

Per cercare nelle pagine man, è possibile usare `/` (ricerca in avanti) o `?` (ricerca indietro) per trovare termini specifici, `n` per passare alla corrispondenza successiva e `q` per uscire. Ad esempio, per cercare informazioni sull'opzione `-r`, digitare `/-r`. Digitare `n` per passare alla corrispondenza successiva.

10. Cosa fa l'opzione -r?

l'opzione `-r` (che sta per **read**) permette a `tcpdump` di **leggere e analizzare i pacchetti da un file salvato** (tipicamente in formato `.pcap` o `.cap`) invece di catturare il traffico in tempo reale da un'interfaccia di rete.

b. Nello stesso terminale, aprire il file di cattura usando il seguente comando per visualizzare i primi 3 pacchetti TCP catturati:

```
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file capture.pcap, link-type EN10MB (Ethernet)
13:58:30.647462 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [S], seq 2432755549, win 29200, options [mss 1460,sackOK,TS val 3864513189 ecr 0,nop,wscale 9], length 0
13:58:30.647543 IP 172.16.0.40.http > 10.0.0.11.58716: Flags [S.], seq 1766419191, ack 2432755550, win 28960, options [mss 1460,sackOK,TS val 58557410 ecr 3864513189,nop,wscale 9], length 0
13:58:30.647544 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 3864513189 ecr 58557410], length 0
```

Per visualizzare l'handshake a 3 vie, potrebbe essere necessario aumentare il numero di righe dopo l'opzione `-c`.

c. Navigare al terminale usato per avviare Mininet. Terminare Mininet inserendo `quit` nella finestra principale del terminale della VM CyberOps.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 2 terms
*** Stopping 5 links
....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@secOps ~]$
```

d. Dopo aver chiuso Mininet, inserire `sudo mn -c` per pulire i processi avviati da Mininet. Inserire la password `cyberops` quando richiesto.

```
[analyst@secOps ~]$ sudo mn -c
[sudo] password for analyst:
```

Domande di Riflessione

1. Ci sono centinaia di filtri disponibili in Wireshark. Una rete di grandi dimensioni potrebbe avere numerosi filtri e molti tipi diversi di traffico. Elenca tre filtri che potrebbero essere utili a un amministratore di rete.

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

1. Ci sono centinaia di filtri disponibili in Wireshark. Una rete di grandi dimensioni potrebbe avere numerosi filtri e molti tipi diversi di traffico. Elenca tre filtri che potrebbero essere utili a un amministratore di rete?

Filtri per http filtri per tcp e indirizzi ip (penso siano questi i 3 più utili)

In reti di grandi dimensioni, i filtri di visualizzazione sono essenziali per isolare i problemi. Ecco tre esempi pratici:

ip.addr == [indirizzo_IP]: Questo è il filtro più comune; permette di vedere tutto il traffico (sia in entrata che in uscita) relativo a un host specifico, isolando le sue comunicazioni dal resto della rete.

tcp.flags.reset == 1: Utilissimo per il troubleshooting. Mostra solo i pacchetti dove una connessione è stata interrotta bruscamente (RST). Se ne vedi molti verso un server, potrebbe esserci un problema di configurazione o un attacco in corso.

http.response.code >= 400: Filtra il traffico web per mostrare solo le risposte di errore (come il classico 404 Not Found o 500 Server Error), permettendo di individuare rapidamente servizi malfunzionanti o tentativi di scansione delle directory.

2. In quali altri modi Wireshark potrebbe essere utilizzato in una rete di produzioni? (non lo so)

Oltre all'analisi dei pacchetti appena fatto nel laboratorio, Wireshark è uno strumento versatile in produzione per:

- ❑ **Risoluzione dei problemi di performance (Troubleshooting):** Identificare colli di bottiglia analizzando i tempi di latenza tra i pacchetti o individuando ritrasmissioni TCP eccessive che indicano una perdita di pacchetti sulla linea.

- ❑ **Analisi forense e Incident Response:** In caso di sospetta compromissione, permette di esaminare i file .pcap (magari estratti proprio con tcpdump -w, l'opposto del comando -r che visto prima) per ricostruire esattamente cosa ha fatto un attaccante o quali dati sono stati esfiltrati.

- ❑ **Verifica della conformità e sicurezza:** Controllare che il traffico sensibile sia effettivamente crittografato