

LABORATORIO 18 FEBBRAIO 2025 S11-L2

Laboratorio:

Utilizzo di Wireshark per Osservare la Stretta di Mano TCP a 3 Vie.

In questo laboratorio, completa i seguenti obiettivi:

- Parte 1: Preparare gli host per catturare il traffico
- Parte 2: Analizzare i pacchetti utilizzando Wireshark
- Parte 3: Visualizzare i pacchetti utilizzando tcpdump

<https://itexamanswers.net/9-2-6-lab-using-wireshark-to-observe-the-tcp-3-way-handshake-answers.html>

9.2.6 Lab – Using Wireshark to Observe the TCP 3-Way Handshake (Answers)

Nel laboratorio odierno il nostro compito sarà analizzare, tramite il tool Wireshark, il 3-Way Handshake TCP. Il requisito richiesto è l'utilizzo della macchina virtuale CyberOps Workstation.

Objectives

- **Part 1: Prepare the Hosts to Capture the Traffic**
- **Part 2: Analyze the Packets using Wireshark**
- **Part 3: View the Packets using tcpdump**

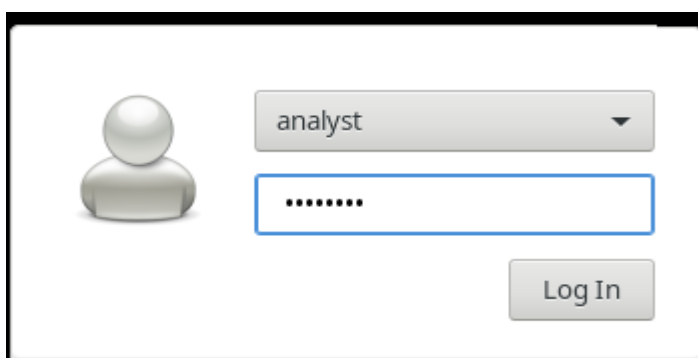
Background / Scenario

In this lab, you will use Wireshark to capture and examine packets generated between the PC browser using the HyperText Transfer Protocol (HTTP) and a web server, such as www.google.com. When an application, such as HTTP or File Transfer Protocol (FTP) first starts on a host, TCP uses the three-way handshake to establish a reliable TCP session between the two hosts. For example, when a PC uses a web browser to surf the internet, a three-way handshake is initiated, and a session is established between the PC host and web server. A PC can have multiple, simultaneous, active TCP sessions with various web sites.

Part 1: Prepare the Hosts to Capture the Traffic

- a. Start the CyberOps VM. Log in with username **analyst** and the password **cyberops**.

PARTE 1: Iniziamo l'esercizio. Il primo passo è avviare la macchina e loggarsi con le credenziali fornite. **User: analyst; pword: cyberops**



Andiamo avanti e ci viene richiesto di avviare Mininet (terminale) ed eseguire i comandi forniti per avviare l'host H1 e H4 e avviare il web server su H4.

b. Start Mininet.

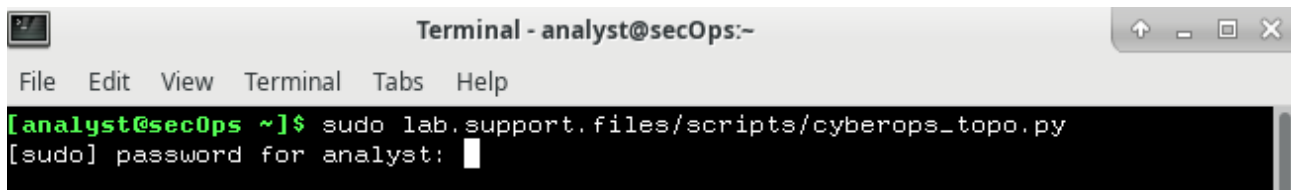
```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
```

c. Start host H1 and H4 in Mininet.

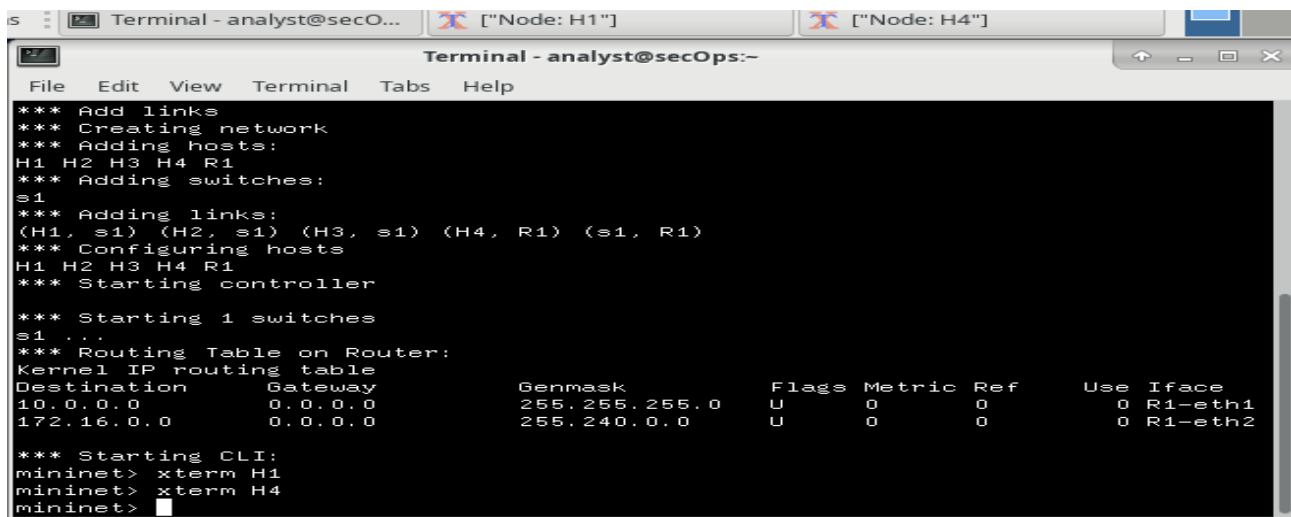
```
*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
```

d. Start the web server on H4.

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_ser
```



A terminal window titled "Terminal - analyst@secOps:-" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [analyst@secOps ~]\$ and the command sudo lab.support.files/scripts/cyberops_topo.py has been entered. The prompt has changed to [sudo] password for analyst: and a cursor is visible.



A terminal window titled "Terminal - analyst@secOps:-" with a menu bar. The output of the script is displayed, showing the creation of a network with hosts H1, H2, H3, H4, and R1, and switch s1. It also shows the configuration of the routing table on the router. The prompt is mininet>.

```
*** Add 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
*** 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

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



A terminal window titled '"Node: H4"' with a menu bar. The prompt is [root@secOps analyst]# and the command /home/analyst/lab.support.files/scripts/reg_server_start.sh has been entered. A cursor is visible at the end of the command.

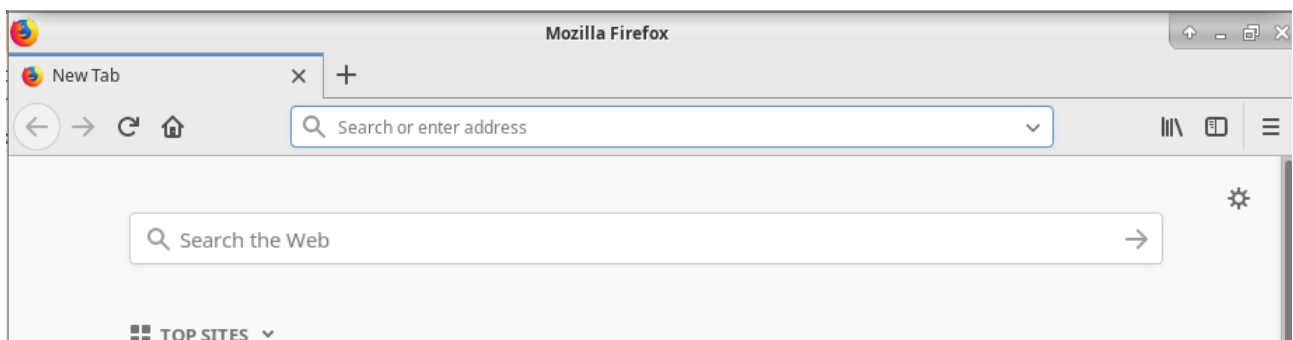
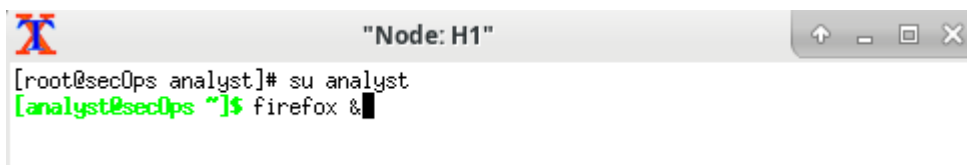
Per motivi di sicurezza, non è possibile eseguire Firefox dall'account utente root. Sull'host H1, utilizzare il comando `switch user` per passare dall'utente root all'account utente analyst e avviamo il browser web su H1.

e. For security purposes, you are not able to run Firefox from the root user account. On host H1, use the `switch user` command to switch from the root user to the analyst user account:

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

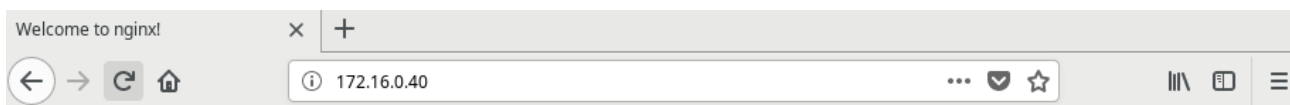
f. Start the web browser on H1. This will take a few moments.

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



A questo punto avviamo una **sessione tcpdump** nel terminale **Node: H1** e inviamo l'output ad un file chiamato **capture.pcap**, con l'opzione **-v** possiamo guardare i progressi. L'acquisizione si interromperà dopo 50 pacchetti in quanto configurata con l'opzione **-c 50**. Dopo l'avvio di `tcpdump`, passare rapidamente a **172.16.0.40** nel browser Web Firefox.

```
[root@secOps analyst]# su analyst
[analyst@secOps ~]$ firefox &
[1] 689
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
[sudo] password for analyst:
tcpdump: listening on H1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
50 packets captured
53 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

PARTE 2: Analizzare i pacchetti con Wireshark.

Apriamo whireshark sul Node: H1, apriamo il file pcap e applichiamo un filtro tcp.

Part 2: Analyze the Packets using Wireshark

Step 1: Apply a filter to the saved capture.

a. Press ENTER to see the prompt. Start Wireshark on **Node: H1**. Click **OK** when prompted by the warning regarding running Wireshark as superuser.

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

b. In Wireshark, click **File > Open**. Select the saved pcap file located at /home/analyst/capture.pcap.

c. Apply a tcp filter to the capture. In this example, the first 3 frames are the interested traffic.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERFECT
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

.Xinitrc	16 bytes	22 Mar 2018
.xsession-errors	3.4 kB	09:51
.xsession-errors.old	374 bytes	16 Nov 2018
capture.pcap	7.0 kB	09:13
Desktop		22 Mar 2018
Downloads		22 Mar 2018

No.	Time	Source	Destination	Protocol	Length	Info
7	1.756462	10.0.0.11	172.16.0.40	TCP	74	34980 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4106187732 TSecr=0 WS=512
8	1.756502	172.16.0.40	10.0.0.11	TCP	74	80 → 34980 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3466838965 TSecr=4106187732 WS=512
9	1.756510	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=4106187732 TSecr=3466838965
10	1.756620	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
11	1.756628	172.16.0.40	10.0.0.11	TCP	66	80 → 34980 [ACK] Seq=1 Ack=312 Win=30208 Len=0 TSval=3466838965 TSecr=4106187732
12	1.757968	172.16.0.40	10.0.0.11	TCP	304	80 → 34980 [PSH, ACK] Seq=1 Ack=312 Win=30208 Len=238 TSval=3466838966 TSecr=4106187732 [TCP segment of a reassembled PDU]
13	1.757974	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=312 Ack=239 Win=30720 Len=0 TSval=4106187733 TSecr=3466838966
14	1.758462	172.16.0.40	10.0.0.11	HTTP	678	HTTP/1.1 200 OK (text/html)
15	1.758467	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=312 Ack=851 Win=31744 Len=0 TSval=4106187734 TSecr=3466838967
22	1.988034	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1
23	1.988272	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
24	1.988491	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=604 Ack=1175 Win=32768 Len=0 TSval=4106187964 TSecr=3466839196
25	2.017709	10.0.0.11	172.16.0.40	HTTP	298	GET /favicon.ico HTTP/1.1
26	2.017859	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
27	2.058560	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=836 Ack=1499 Win=34304 Len=0 TSval=4106188034 TSecr=3466839226

A questo punto vanno esaminate le informazioni interne ai pacchetti, come indirizzi IP, i numeri di porta TCP e i flag di controllo TCP. Vanno individuate le porte di origine e di destinazione e il flag impostato sul frame 1.

Step 2: Examine the information within packets including IP addresses, TCP port numbers, and TCP control flags.

- In this example, frame 1 is the start of the three-way handshake between the PC and the server on H4. In the packet list pane (top section of the main window), select the first packet, if necessary.
- Click the **arrow** to the left of the Transmission Control Protocol in the packet details pane to expand it and examine the TCP information. Locate the source and destination port information.
- Click the **arrow** to the left of the Flags. A value of 1 means that flag is set. Locate the flag that is set in this packet.

Note: You may have to adjust the top and middle windows sizes within Wireshark to display the necessary information

No.	Time	Source	Destination	Protocol	Length	Info
7	1.756462	10.0.0.11	172.16.0.40	TCP	74	34980 → 80 [SYN] Seq=0 Win=2920
8	1.756502	172.16.0.40	10.0.0.11	TCP	74	80 → 34980 [SYN, ACK] Seq=0 Ack=
9	1.756510	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=1 Ack=1 Win
10	1.756620	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
▶ Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)						
▶ Ethernet II, Src: 02:88:6a:7f:9d:a5 (02:88:6a:7f:9d:a5), Dst: 76:d2:7e:e8:03:d1 (76:d2:7e:e8:03:d1)						
▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40						
▼ Transmission Control Protocol, Src Port: 34980, Dst Port: 80, Seq: 0, Len: 0						
Source Port: 34980						
Destination Port: 80						
[Stream index: 0]						
[TCP Segment Len: 0]						
Sequence number: 0 (relative sequence number)						
[Next sequence number: 0 (relative sequence number)]						
Acknowledgment number: 0						
1010 = Header Length: 40 bytes (10)						
▶ Flags: 0x002 (SYN)						
Window size value: 29200						
[Calculated window size: 29200]						
Checksum: 0xb671 [unverified]						
[Checksum Status: Unverified]						
Urgent pointer: 0						
▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale						
▶ [Timestamps]						
▼ Flags: 0x002 (SYN)						
000. = Reserved: Not set						
...0 = Nonce: Not set						
.... 0... = Congestion Window Reduced (CWR): 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						

Rispondere alle seguenti domande:

Il numero di porta di origine TCP è? **34980**

Come classifichereesti la porta di origine? **DINAMICA**

Qual è il numero di porta di destinazione TCP? **80**

Come classifichereesti la porta di destinazione? **HTTP**

Quale flag è impostato? **SYN FLAG**

Su cosa è impostato il numero di sequenza relativo? **0**

d. Select the next packet in the three-way handshake. In this example, this is frame 2. This is the web server replying to the initial request to start a session.

Procedere con l'analisi del secondo frame che corrisponde alla risposta del server alla richiesta di iniziare la sessione da parte del client e rispondere alle seguenti domande:

Quali sono i valori delle porte di origine e di destinazione? **SRC Port=80; DST Port=34980**

Quali flag sono impostati? **SYN E ACK**

Su cosa sono impostati i numeri di sequenza e di riconoscimento relativi? **0 E 1**

8	1.756502	172.16.0.40	10.0.0.11	TCP	74	80 → 34980 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3466838965 TSecr=4106187732 WS=512
9	1.756510	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=4106187732 TSecr=3466838965
10	1.756620	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1

▼ Transmission Control Protocol, Src Port: 80, Dst Port: 34980, Seq: 0, Ack: 1, Len: 0

Source Port: 80
Destination Port: 34980
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1010 = Header Length: 40 bytes (10)

▼ Flags: 0x012 (SYN, ACK)

000. = Reserved: Not set
...0 = Nonce: Not set
....0... = Congestion Window Reduced (CWR): 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
....0... = Fin: Not set
[TCP Flags:A..S]

Concludere analizzando il terzo pacchetto che conclude l'handshake e rispondere alle domande.

9	1.756510	10.0.0.11	172.16.0.40	TCP	66	34980 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=4106187732 TSecr=3466838965
10	1.756620	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1

▶ Ethernet II, Src: 02:88:6a:7f:9d:a5 (02:88:6a:7f:9d:a5), Dst: 76:d2:7e:e8:03:d1 (76:d2:7e:e8:03:d1)

▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40

▼ Transmission Control Protocol, Src Port: 34980, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 34980
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 = Header Length: 32 bytes (8)
▼ Flags: 0x010 (ACK)
000. = Reserved: Not set
...0 = Nonce: Not set
....0... = Congestion Window Reduced (CWR): Not set
....0... = ECN-Echo: Not set
....0... = Urgent: Not set
....1... = Acknowledgment: Set
....0... = Push: Not set
....0... = Reset: Not set
....0... = Syn: Not set

Quale flag è impostato? **ACK**

I numeri di sequenza e di riconoscimento relativi sono impostati su 1 come punto di partenza. Viene stabilita la connessione TCP e può iniziare la comunicazione tra il computer di origine e il server Web.

PARTE 3: Visualizzare i pacchetti utilizzando tcpdump.

Part 3: View the packets using tcpdump

You can also view the pcap file and filter for the desired information.

a. Open a new terminal window, enter `man tcpdump`. **Note:** You may need to press ENTER to see the prompt.

Apriamo un nuovo terminale e diamo il comando **man tcpdump**.

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
TCPDUMP(1)          General Commands Manual          TCPDUMP(1)

NAME
  tcpdump - dump traffic on a network

SYNOPSIS
  tcpdump [ -AbDefhHIJKlLnMOpqStuUvxX# ] [ -B buffer_size ]
  [ -c count ]
  [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
  [ -i interface ] [ -j timestamp_type ] [ -m module ] [ -M secret ]
  [ --number ] [ -Q inout|inout ]
  [ -r file ] [ -U file ] [ -s snaplen ] [ -T type ] [ -w file ]
```

All'interno del manuale dobbiamo capire a cosa corrisponde il comando -r. Per farlo possiamo muoverci utilizzando comandi specifici. Utilizziamo quello consigliato ovvero **type /-r**.

To search through the man pages, you can use / (searching forward) or ? (searching backward) to find specific terms, and n to forward to the next match and q to quit. For example, search for the information on the switch -r, type /-r. Type n to move to the next match.

What does the switch -r do?

```
/-r
```

```
-r file
Read packets from file (which was created with the -w option or
by other tools that write pcap or pcap-ng files). Standard
input is used if file is '-'.

```

Quesito: cosa fa -r? L'opzione -r consente di leggere il pacchetto dal file che è stato salvato utilizzando l'opzione -w o altri strumenti che scrivono file pcap o pcap-ng.

A questo punto, nello stesso terminale, apriamo il file di acquisizione usando il comando fornito per visualizzare i primi 3 pacchetti TCP ottenuti.

b. In the same terminal, open the capture file using the following command to view the first 3 TCP packets captured:

```
[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.ethercat > 172.16.0.40.http: Flags [S], seq
13:58:30.647543 IP 172.16.0.40.http > 10.0.0.11.ethercat: Flags [S.], seq
13:58:30.647544 IP 10.0.0.11.ethercat > 172.16.0.40.http: Flags [.], ack
```

To view the 3-way handshake, you may need to increase the number of lines after the -c option.

```
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
09:13:30.791625 IP 10.0.0.11.ethercat > 172.16.0.40.http: Flags [S], seq 3944116733, win 29200, options [mss
1460,sackOK,TS val 4106187732 ecr 0,nop,wscale 9], length 0
09:13:30.791665 IP 172.16.0.40.http > 10.0.0.11.ethercat: Flags [S.], seq 3699546395, ack 3944116734, win 289
60, options [mss 1460,sackOK,TS val 3466838965 ecr 4106187732,nop,wscale 9], length 0
09:13:30.791673 IP 10.0.0.11.ethercat > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 4
106187732 ecr 3466838965], length 0
```

Passiamo al terminale utilizzato per avviare Minimet e chiudiamolo con il comando **quit**. Dopo averlo chiuso inseriamo **sudo mn -c** per ripulire i processi avviati da Minimet.

d. After quitting Mininet, enter `sudo mn -c` to clean up the processes started by Mininet. Enter the password **cyberops** when prompted.

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

```
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 ~]$
```

