

# Laboratori giorno 2 – Cisco CyberOps

## Analisi di stretta di mano a 3 vie

Utilizzeremo 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

### Parte 1

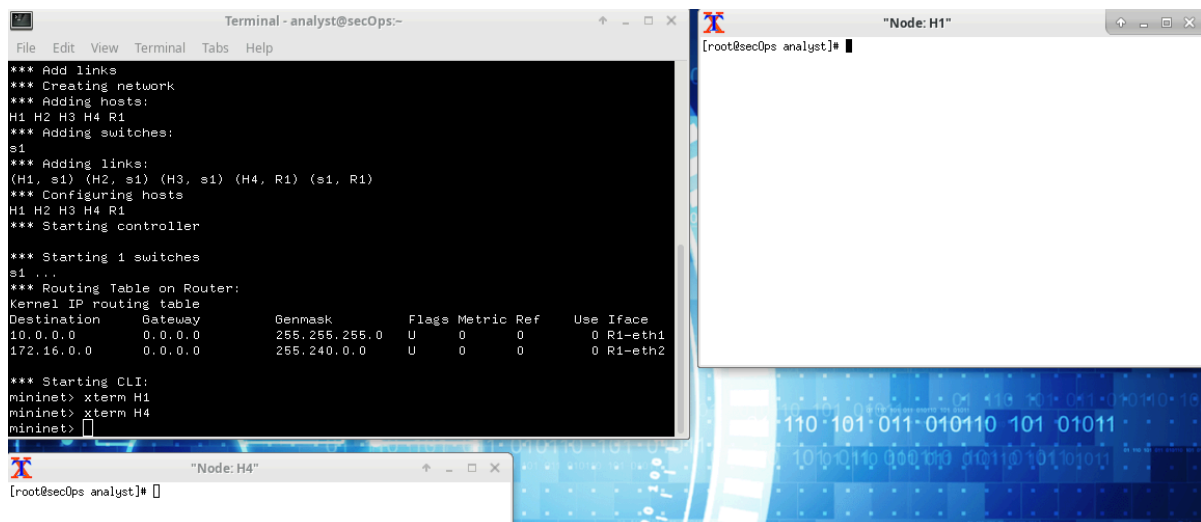
Avvia la VM CyberOps. Accedi con il nome utente `analyst` e la password `cyberops`.

Avvia Mininet da terminale con il comando `sudo lab.support.files/scripts/cyberops_topo.py`.

**Mininet** è un emulatore di rete open-source che consente di creare, configurare e simulare reti virtuali su un singolo computer. Viene utilizzato principalmente per sperimentare protocolli, testare applicazioni di rete e studiare il comportamento di reti SDN (Software-Defined Networking).

```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
-----  
*** 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> █
```

Avvia gli host H1 e H4 che verranno configurati automaticamente da Mininet con i comandi `xterm H1` e `xterm H4`.



```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
*** 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> █  
  
"Node: H1"  
[root@secOps analyst]# █  
  
"Node: H4"  
[root@secOps analyst]# █
```

Avvia il server web sul terminale H4 con il comando `/home/analyst/lab.support.files/scripts/reg_server_start.sh`.

```
"Node: H4"
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start
.sh
[root@secOps analyst]#
```

Sull'host H1, dal terminale H1, passiamo da utente root a utente analyst con il comando `su analyst`.

Eseguiamo poi firefox con il comando `firefox &`.

```
"Node: H1"
[root@secOps analyst]# su analyst
[analyst@secOps ~]$ firefox &
[1] 937
```

Sempre da terminale H1, avviamo una sessione tcpdump e generiamo un output tramite un file chiamato capture.pcap.

Il comando è `sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analista/capture.pcap`.

Con l'opzione -v, puoi 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.p
cap
tcpdump: listening on H1-eth0, link-type EN10MB (Ethernet), capture size 262144
bytes
50 packets captured
52 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```

## **Parte 2**

Apri il file appena generato con Wireshark. Il file pcap salvato si trova in `/home/analyst/capture.pcap`.

Applica un filtro `tcp` alla cattura. I primi 3 pacchetti sono quelli che ci interessano, dato che riguardano la stretta di mano a 3 vie.

capture.pcap [Wireshark 2.5.1]						
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help						
Filter: tcp Expression... Clear Apply Save						
No.	Time	Source	Destination	Protocol	Length	Info
6	1.406277	10.0.0.11	172.16.0.40	TCP	74	60308 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=457711534 TSecr=0 WS=512
7	1.406325	172.16.0.40	10.0.0.11	TCP	74	80 → 60308 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3475856124 TSecr=457711534 WS=512
8	1.406334	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=457711534 TSecr=3475856124
9	1.406445	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
10	1.406454	172.16.0.40	10.0.0.11	TCP	66	80 → 60308 [ACK] Seq=1 Ack=312 Win=30208 Len=0 TSval=3475856124 TSecr=457711534
11	1.417277	172.16.0.40	10.0.0.11	TCP	304	80 → 60308 [PSH, ACK] Seq=1 Ack=312 Win=30208 Len=238 TSval=3475856135 TSecr=457711534 [TCP segment of a reassembled
12	1.417287	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=312 Ack=239 Win=30720 Len=0 TSval=457711545 TSecr=3475856135
13	1.418084	172.16.0.40	10.0.0.11	HTTP	678	HTTP/1.1 200 OK (text/html)
14	1.418091	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=312 Ack=851 Win=31744 Len=0 TSval=457711545 TSecr=3475856135
19	1.472437	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1
20	1.472569	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
21	1.472782	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=604 Ack=1175 Win=32768 Len=0 TSval=457711600 TSecr=3475856190
44	11.503599	10.0.0.11	172.16.0.40	TCP	66	[TCP Keep-Alive] 60308 → 80 [ACK] Seq=603 Ack=1175 Win=32768 Len=0 TSval=457721631 TSecr=3475856190
45	11.503638	172.16.0.40	10.0.0.11	TCP	66	[TCP Keep-Alive ACK] 80 → 60308 [ACK] Seq=1175 Ack=604 Win=31232 Len=0 TSval=3475866221 TSecr=457711600
50	13.512981	10.0.0.11	172.16.0.40	HTTP	484	GET / HTTP/1.1

## Analizziamo il primo pacchetto SYN:

No.	Time	Source	Destination	Protocol	Length	Info
6	1.406277	10.0.0.11	172.16.0.40	TCP	74	60308 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=457711534 TSecr=0 WS=512
7	1.406325	172.16.0.40	10.0.0.11	TCP	74	80 → 60308 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3475856124 TSecr=457711534 WS=512
8	1.406334	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=457711534 TSecr=3475856124
9	1.406445	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
▶ Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) ▶ Ethernet II, Src: 1a:0e:9a:04:81:94 (1a:0e:9a:04:81:94), Dst: de:5e:73:93:ce:f0 (de:5e:73:93:ce:f0) ▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40 ▼ Transmission Control Protocol, Src Port: 60308, Dst Port: 80, Seq: 0, Len: 0 Source Port: 60308 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

▶ [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]

Da questo log possiamo notare che:

- La porta sorgente è **58716**, una porta dinamica o privata.
- La porta di destinazione è la porta **80**, la porta nota HTTP.
- La bandiera è di tipo **SYN**.

## Analizziamo il secondo pacchetto SYN /ACK:

No.	Time	Source	Destination	Protocol	Length	Info
6	1.406277	10.0.0.11	172.16.0.40	TCP	74	60308 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=457711534 TSecr=0 WS=512
7	1.406325	172.16.0.40	10.0.0.11	TCP	74	80 → 60308 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3475856124 TSecr=0
8	1.406334	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=457711534 TSecr=3475856124
<b>Transmission Control Protocol, Src Port: 80, Dst Port: 60308, Seq: 0, Ack: 1, Len: 0</b> Source Port: 80 Destination Port: 60308 [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)						
<b>Flags: 0x012 (SYN, ACK)</b> 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...1. = Syn: Set ▶ [Expert Info (Chat/Sequence): Connection establish acknowledge (SYN+ACK): server port 80]						

Da questo pacchetto

- La porta di origine è ora **80**.
- La porta di destinazione è ora **58716**.
- Ci sono due flag: una di riconoscimento (**ACK**) e il flag Syn (**SYN**).

## Analizziamo il terzo pacchetto ACK:

No.	Time	Source	Destination	Protocol	Length	Info
6	1.406277	10.0.0.11	172.16.0.40	TCP	74	60308 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=457711534 TSecr=0 WS=512
7	1.406325	172.16.0.40	10.0.0.11	TCP	74	80 → 60308 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3475856124 TSecr=
8	1.406334	10.0.0.11	172.16.0.40	TCP	66	60308 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=457711534 TSecr=3475856124

▼ Transmission Control Protocol, Src Port: 60308, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 60308

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

....0 = Fin: Not set

TCP Flags: .....A.....1

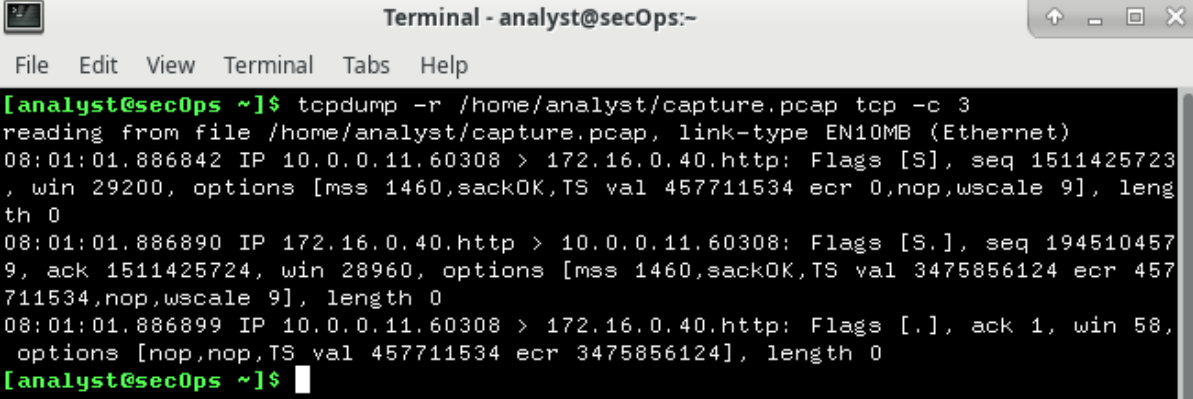
Dal log di questo pacchetto notiamo che:

- È presente la flag di conferma (ACK).

## **Parte 3**

Possiamo analizzare il log catturato anche tramite il terminale con il comando `tcpdump -r /home/analista/capture.pcap tcp -c 3`.

Suggerisco comunque di utilizzare wireshark per avere una visione migliore dei pacchetti.



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
08:01:01.886842 IP 10.0.0.11.60308 > 172.16.0.40.http: Flags [S], seq 1511425723, win 29200, options [mss 1460,sackOK,TS val 457711534 ecr 0,nop,wscale 9], length 0
08:01:01.886890 IP 172.16.0.40.http > 10.0.0.11.60308: Flags [S.], seq 1945104579, ack 1511425724, win 28960, options [mss 1460,sackOK,TS val 3475856124 ecr 457711534,nop,wscale 9], length 0
08:01:01.886899 IP 10.0.0.11.60308 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 457711534 ecr 3475856124], length 0
[analyst@secOps ~]$
```