

# Usare Wireshark per Osservare l'Handshake a 3 Vie TCP

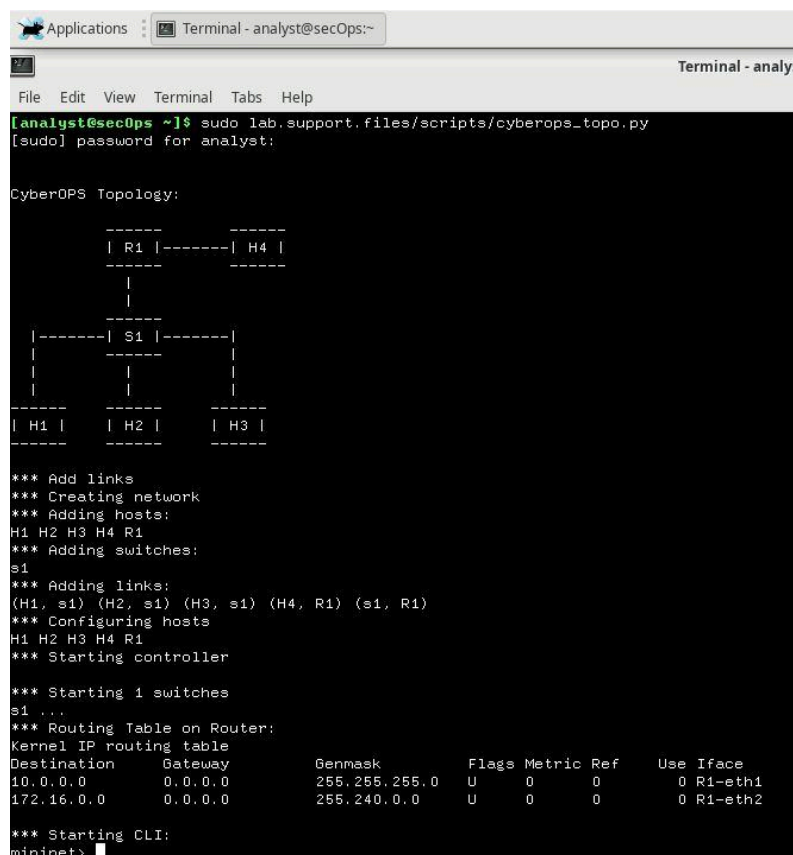
## Obiettivi

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

**Macchine utilizzate:** CyberOps Workstation

## Parte 1: Preparare gli Host per Catturare il Traffico

### Avvio di Mininet



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

CyberOPS Topology:

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

*** 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>
```

Mininet è stato avviato grazie al comando **sudo lab.support.files/scripts/cyberops\_topo.py**

Successivamente, sono stati avviati gli Host **H1** & **H4** su Mininet con l'utilizzo dei rispettivi comandi **xterm H1** e **xterm H4**.



Su H4 è stato avviato il Web Server con l'utilizzo del comando **/home/analyst/lab.support.files/scripts/reg\_server\_start.sh**

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start
.sh
[root@secOps analyst]# su analyst
[analyst@secOps ~]$ firefox &
[1] 860
```

Quindi si è passati all'utente **analyst** con il comando **su** e si è aperto Firefox con **firefox &**.

Grazie al comando **sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap** sull'Host **H1** si è aperta una sessione **tcpdump**, inviando l'output ad un file chiamato **capture.pcap**.

Dopo l'avvio di **tcpdump**, si è visitato **172.16.0.40** nel browser web Firefox.

Successivamente, dopo aver ricevuto i 50 pacchetti indicati con l'opzione **-c** è stato aperto **Wireshark** con il comando **wireshark-gtk &** sempre sull'Host **H1**.

Nel percorso indicato nel comando utilizzato per **tcpdump** visto prima, è stato creato un file, salvato come **capture.pcap**.

**Per osservare correttamente l'Handshake a 3 vie con i pacchetti TCP è stato applicato il filtro TCP su Wireshark.**

| No. | Time     | Source      | Destination | Protocol | Length | Info                                     |
|-----|----------|-------------|-------------|----------|--------|--|
| 23  | 7.597075 | 10.0.0.11   | 172.16.0.40 | TCP      | 74     | 57722 → 80 [SYN] Seq=0 Win=29200 Len=0   |
| 24  | 7.597100 | 172.16.0.40 | 10.0.0.11   | TCP      | 74     | 80 → 57722 [SYN, ACK] Seq=0 Ack=1 Win=28 |
| 25  | 7.597106 | 10.0.0.11   | 172.16.0.40 | TCP      | 66     | 57722 → 80 [ACK] Seq=1 Ack=1 Win=29696   |
| 26  | 7.597162 | 10.0.0.11   | 172.16.0.40 | HTTP     | 358    | GET /favicon.ico HTTP/1.1                |
| 27  | 7.597168 | 172.16.0.40 | 10.0.0.11   | TCP      | 66     | 80 → 57722 [ACK] Seq=1 Ack=293 Win=3020  |

Cliccando su Transmission Control Protocol in basso, è possibile vedere le informazioni TCP.

|  |
|--|
| ▼ Transmission Control Protocol, Src Port: 57722, Dst Port: 80, Seq: 0, Len: 0 |
| Source Port: 57722   |
| Destination Port: 80   |

In questo caso la **porta d'origine** è 57722 mentre quella di **destinazione** è la porta 80.

Facendo click invece su Flags, si è potuto localizzare il flag impostato.

|  |
|--|
| ▼ 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                          |

## **Pacchetto 1.**

**Richiesta iniziale dell'handshake a 3 vie.**

### **Quesiti e Risposte:**

**Qual è il numero di porta TCP di origine?**

La porta TCP di origine è la porta 57722.

**Come classificheresti la porta di origine?**

La porta di origine è classificabile come porta privata.

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

La porta TCP di destinazione è la porta 80.

**Come classificheresti la porta di destinazione?**

La porta di destinazione è classificabile come porta utente o registrata, utilizzata per servizi o applicazioni. In particolare la porta 80 viene utilizzata da server web, quindi **HTTP**.

**Quale flag è impostato?**

E' impostato il flag **SYN**.

**A quale valore è impostato il numero di sequenza relativo?**

Sequence number: 0 (relative sequence number)

Il valore è impostato a 0.

## **Pacchetto 2.**

**Risposta del Server Web alla richiesta iniziale.**

### **Quesiti e Risposte:**

**Quali sono i valori delle porte di origine e destinazione?**

I valori delle porte di origine e di destinazione sono inversi a quelli visti in precedenza. La porta di origine infatti è la porta 80 e la porta di destinazione è la 57722.

Quali flag sono impostati?

I Flag impostati sono **SYN** e **Acknowledgment**.

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

I valori impostati dei numeri relativi di sequenza e acknowledgment sono entrambi 0.

**Pacchetto 3.**

Il Flag in questo pacchetto è impostato solo su Acknowledgment.

**Il processo è quindi Pacchetto 1 SYN -> Pacchetto 2 SYN ACK -> Pacchetto 3 ACK.**

**Utilizzo di tcpdump.**

```
it to save the packet data to a file for later analysis, and/or with the -r flag, which causes it to  
-r file
```

L'opzione **-r** legge i pacchetti di un determinato file.

**Quesiti e Risposte**

**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) Un amministratore di una rete di grandi dimensioni potrebbe avere la necessità primaria di dover isolare il traffico il più possibile, quindi di conseguenza sarebbero utili:

- **Filtro per Indirizzo IP specifico** usato con **ip.addr == <ip>**  
Permette di monitorare tutto il traffico da o verso un host specifico.
- **Filtro per tipo di protocollo (DNS, HTTP, TCP)**, inserendo uno di questi protocolli come filtro potrebbe isolare il traffico di un determinato protocollo.
- **Filtro per traffico ARP anomalo o in eccesso**, inserendo arp come filtro, permette di rilevare problemi di risoluzione degli indirizzi **MAC/IP**, o di individuare attacchi come **ARP spoofing o flooding**.

2) Wireshark è uno strumento molto versatile, può avere altri utilizzi oltre l'analisi dei pacchetti.

Questi possono essere **la diagnosi di problemi di connettività, l'analisi delle prestazioni della rete, l'individuazione di anomalie e attacchi, il debug di applicazioni e servizi, la verifica della sicurezza e conformità.**