

## REPORT 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

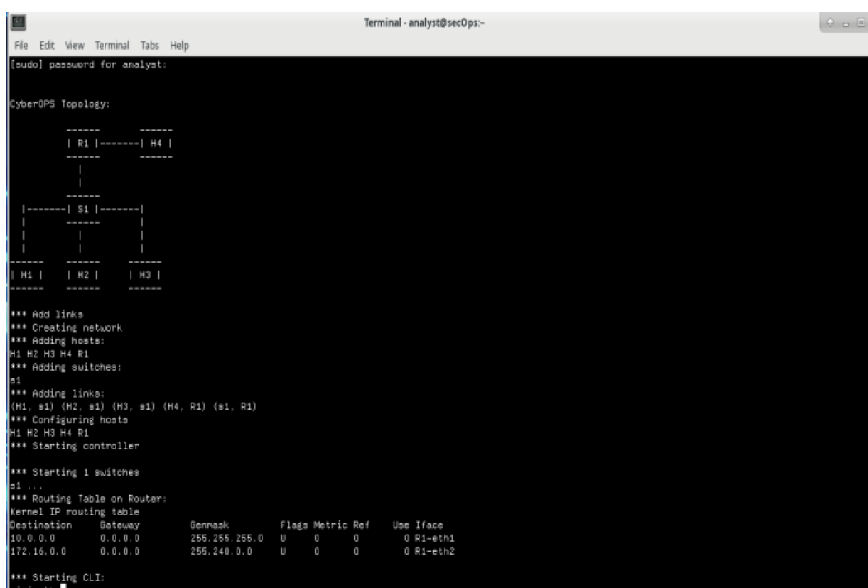
## SOLUZIONE

---

In questo laboratorio utilizzerò Wireshark per catturare ed esaminare i pacchetti generati tra un browser mediante protocollo HTTP e un web server, per esempio [www.google.com](http://www.google.com). Nel momento in cui un'applicazione, http o ftp, inizia una comunicazione con un host si stabilisce una sessione tcp tra gli host utilizzando una three-way-handshake. Per esempio quando un browser naviga in rete si stabilisce una three way handshake tra il pc e il web-server. Un pc può avere più sessioni TCP aperte contemporaneamente.

Per prima cosa effettuo il login alla macchina Cyberops Workstation con username **Analyst** e password **cyberops**. E avvio Mininet con il comando:

**[analyst@secOps ~]\$ sudo lab.support.files/scripts/cyberops\_topo.py**



```
Terminal - analyst@secOps~
File Edit View Terminal Tabs Help

[cloud] 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
0.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 S1-eth1
172.16.0.0 0.0.0.0 255.240.0.0 U 0 0 0 R1-eth2

*** Starting CLI:
mininet>
```

Avvio poi l'host H1 e H4 su Mininet:

**\* Starting CLI:**

**mininet> xterm H1**

**mininet> xterm H4**

Avvio ora il web server su H4:

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

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

Per motivi di sicurezza non è possibile avviare Firefox dal account di root. Passo quindi all' account **analyst**:

```
su analyst
```



Avvio quindi firefox

**Firefox &**

A questo punto avvio una sessione **tcpdump** nel terminale **Node: H1** e invia l' output ad un file chiamato **capture.pcap**. Con l' opzione **-v** sarà possibile osservare i progressi. La cattura si fermerà dopo 50 pacchetti con l' opzione **-c 50**.

```
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w
```

```
/home/analyst/capture.pcap
```

Procedo quindi con l' analisi su **wireshark**:

**wireshark &**

Apro il file **capture.pcap**

Applico il filtro TCP.

Filter:		tcp	Expression... Clear Apply Save			
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_PE
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=1
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=3864
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

Procedo quindi andando ad esaminare le informazione nei pacchetti, gli indirizzi IP, i numeri di porta, e le flag di controllo.

La porta Tcp sorgente è la 58716, la sua classificazione sarà Dinamica o privata.

La porta Destinazione è la porta 80, che come sappiamo è una di quelle note destinate al protocollo HTTP. Si tratta di una SYN con numero di sequenza 0.

Procediamo poi con il pacchetto 2, in questo caso avremo che la porta sorgente sarà la 80 e la destinataria la 58716, esattamente l' inverso rispetto a prima. Si tratta di una SYN, ACK avremo

numero di sequenza 0 e numero di acknowledgement 1. Abbiamo infine il terzo pacchetto nel quale avremo flag ACK e numero di sequenza e acknowledgement a 1. Abbiamo quindi stabilito una sessione TCP tra Pc e server.