

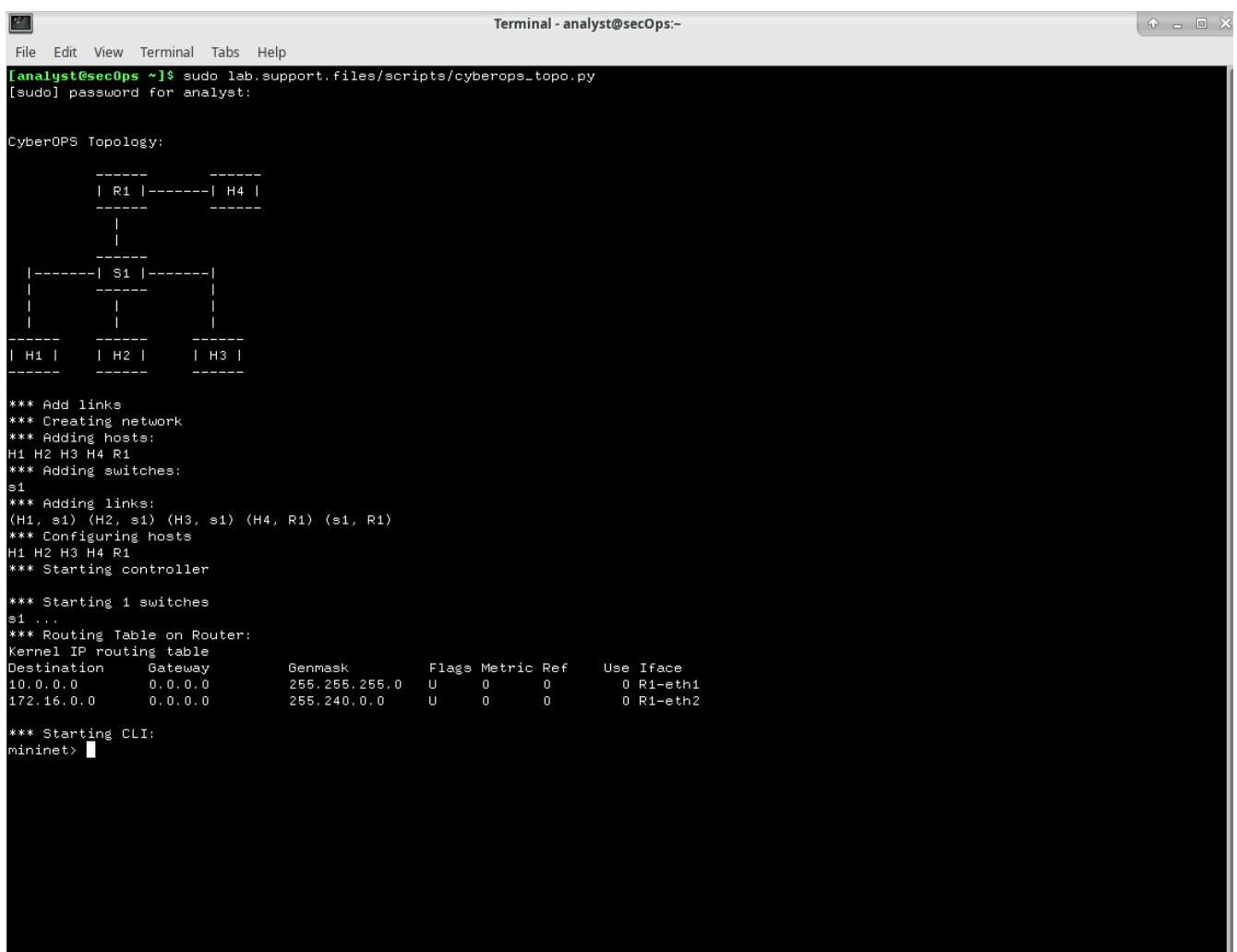
# S11L3 - Analisi del Traffico di Rete con Wireshark e tcpdump

## Analisi del Traffico di Rete con Wireshark e tcpdump

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

- Avvia la VM CyberOps. Accedi con nome utente analyst e la password cyberops.
- Avvia Mininet.

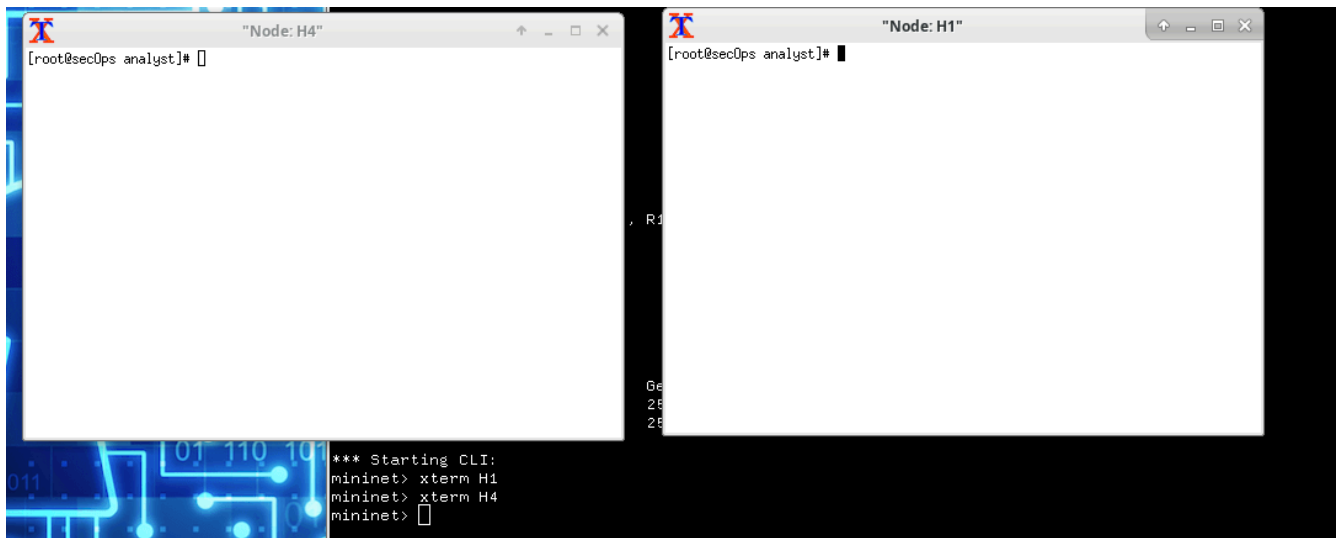
```
sudo lab.support.files/scripts/cyberops_topo.py
```



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

- Avvia gli host H1 e H4 in Mininet.

```
mininet> xterm H1  
mininet> xterm H4
```



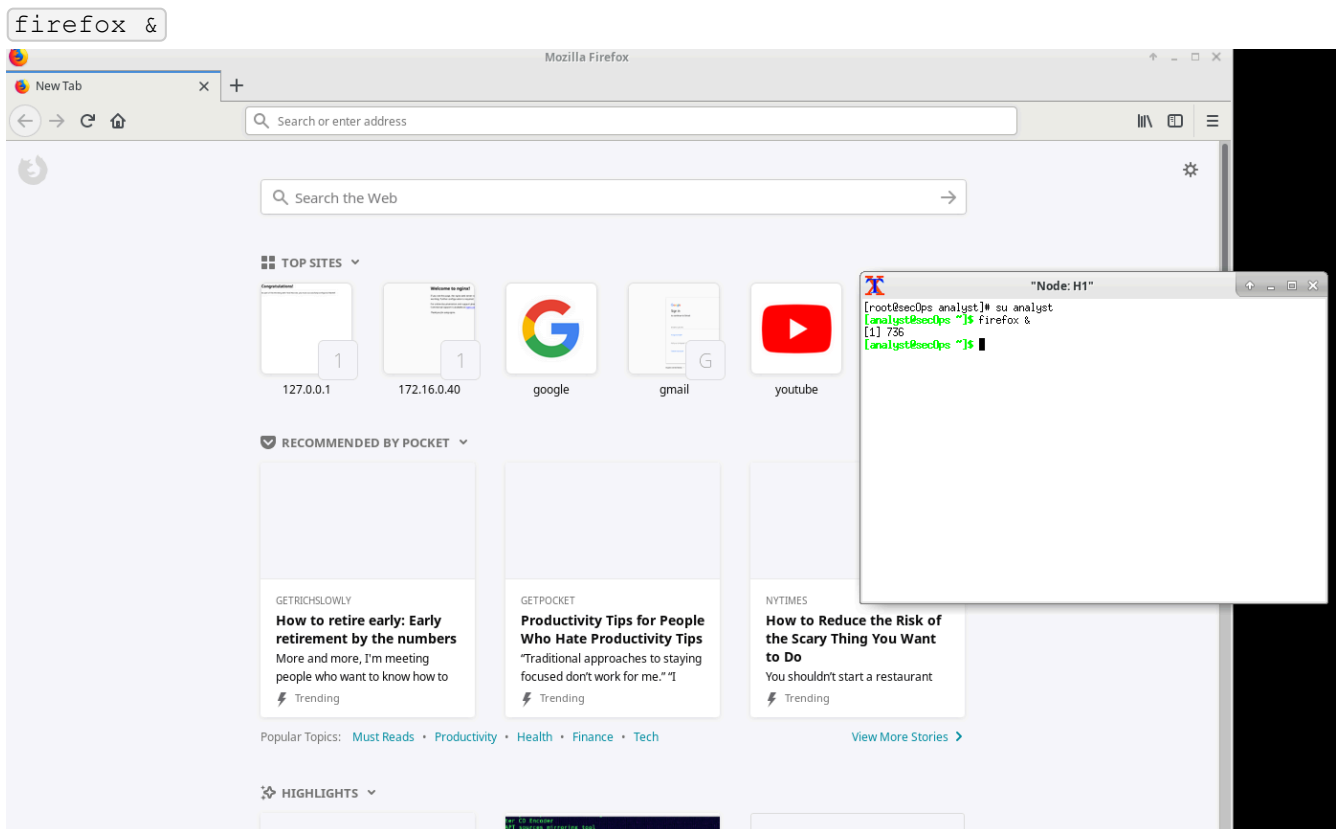
- **Avvia il server web su H4.**

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

- 
- **Passo all'account utente analyst su H1 usando il comando su:**

```
su analyst
```

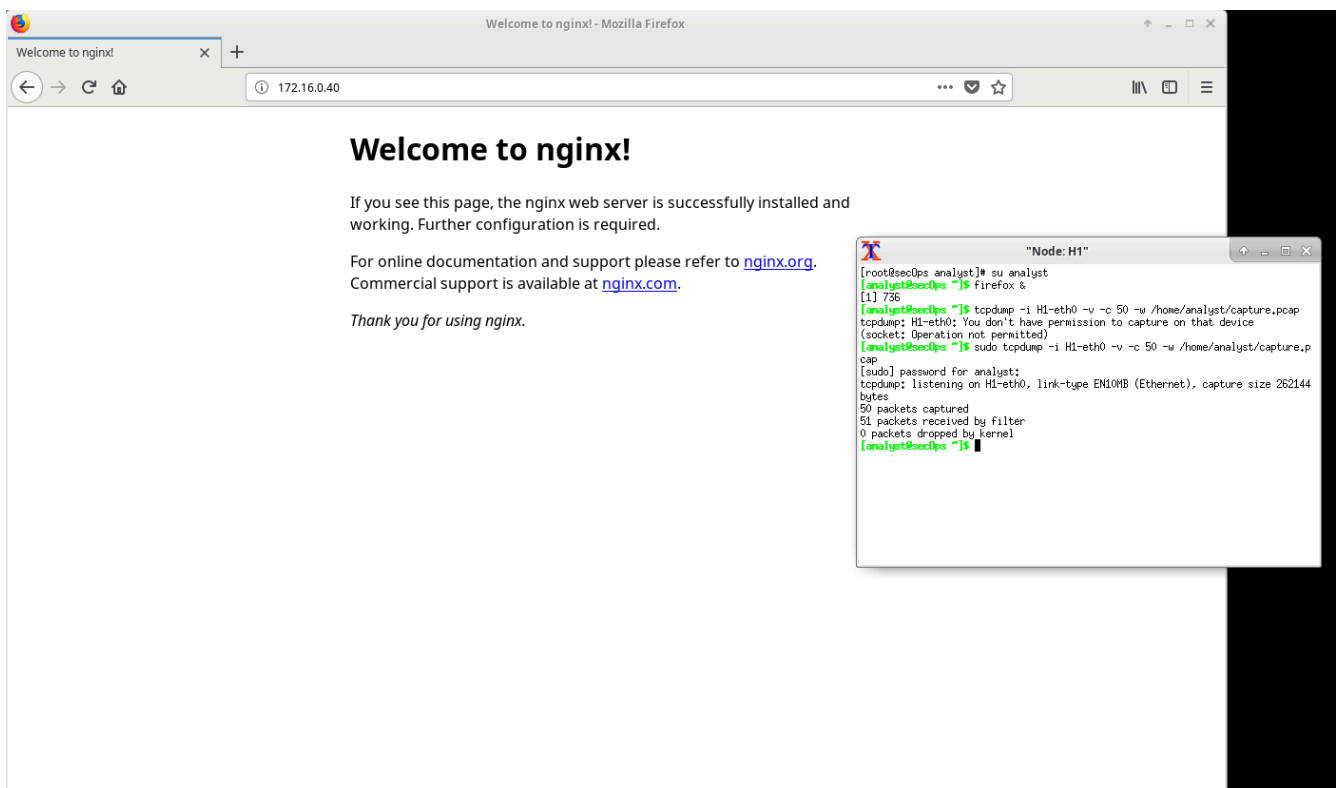
- **Avvia il browser Firefox su H1. Ci vorranno alcuni momenti.**



- Dopo l'apertura della finestra di Firefox, avvio una sessione tcpdump nel terminale di H1 per catturare i pacchetti e inviare l'output a un file chiamato capture.pcap. Con l'opzione -v, posso monitorare il progresso. Questa cattura si fermerà dopo aver catturato 50 pacchetti se uso -c 50.

```
sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
```

- Dopo l'avvio di tcpdump, vado su 172.16.0.40 nel browser Firefox.

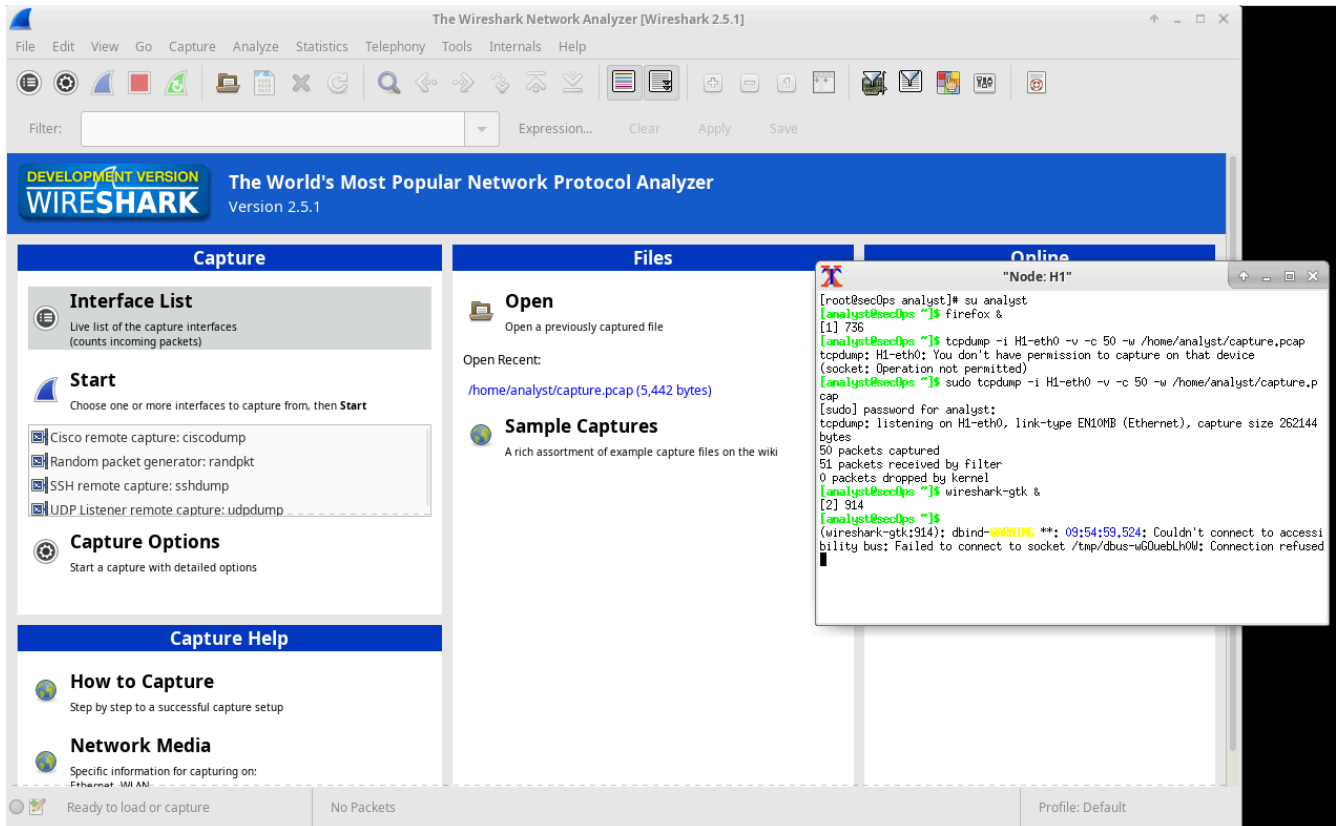


## Parte 2: Analizzare i Pacchetti con Wireshark

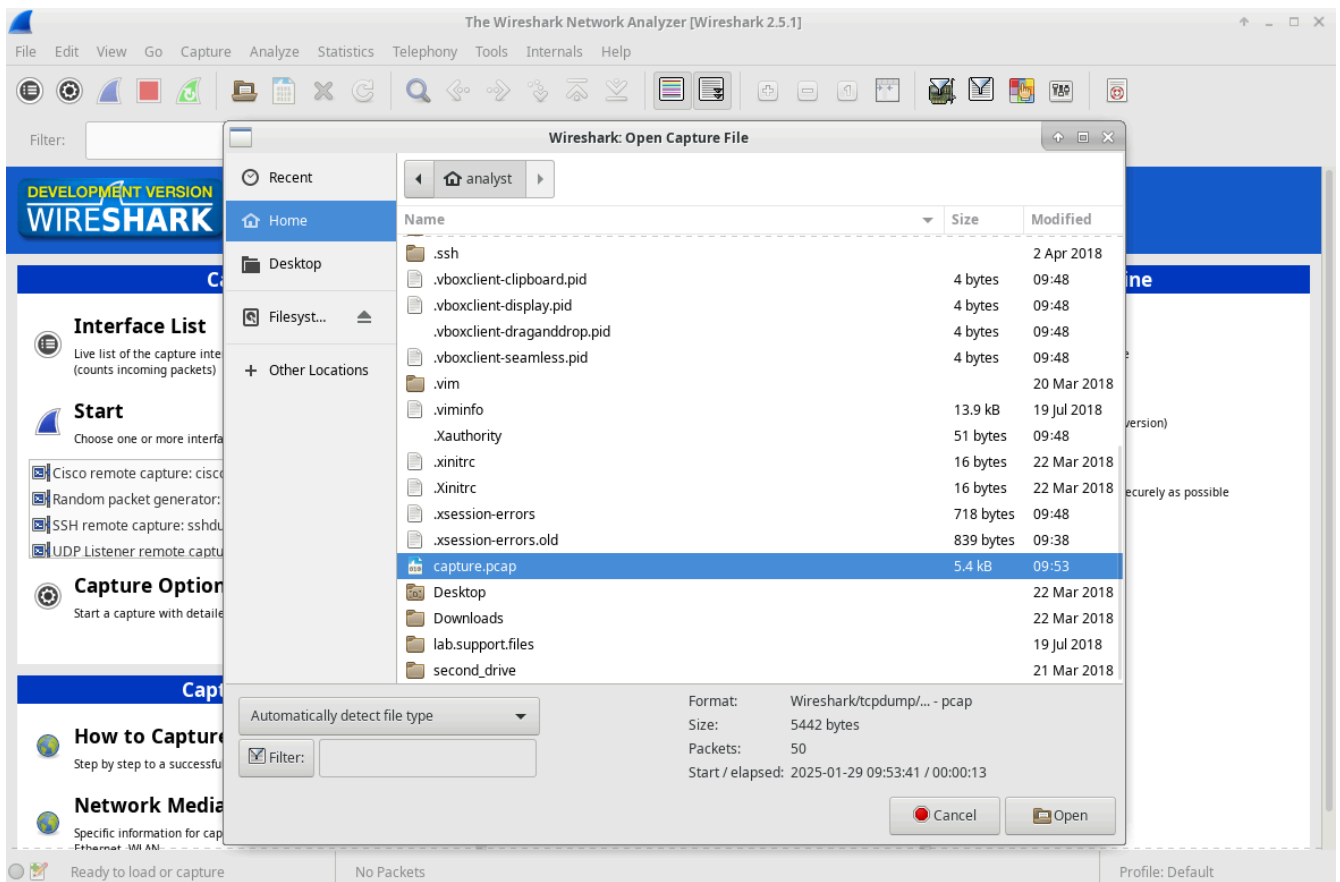
# STEP 1

- Avvio Wireshark su H1.

```
wireshark-gtk &
```



- In Wireshark, clicca su File > Open e seleziono il file capture.pcap salvato in /home/analyst/capture.pcap.



- Applico un filtro tcp sui pacchetti catturati.

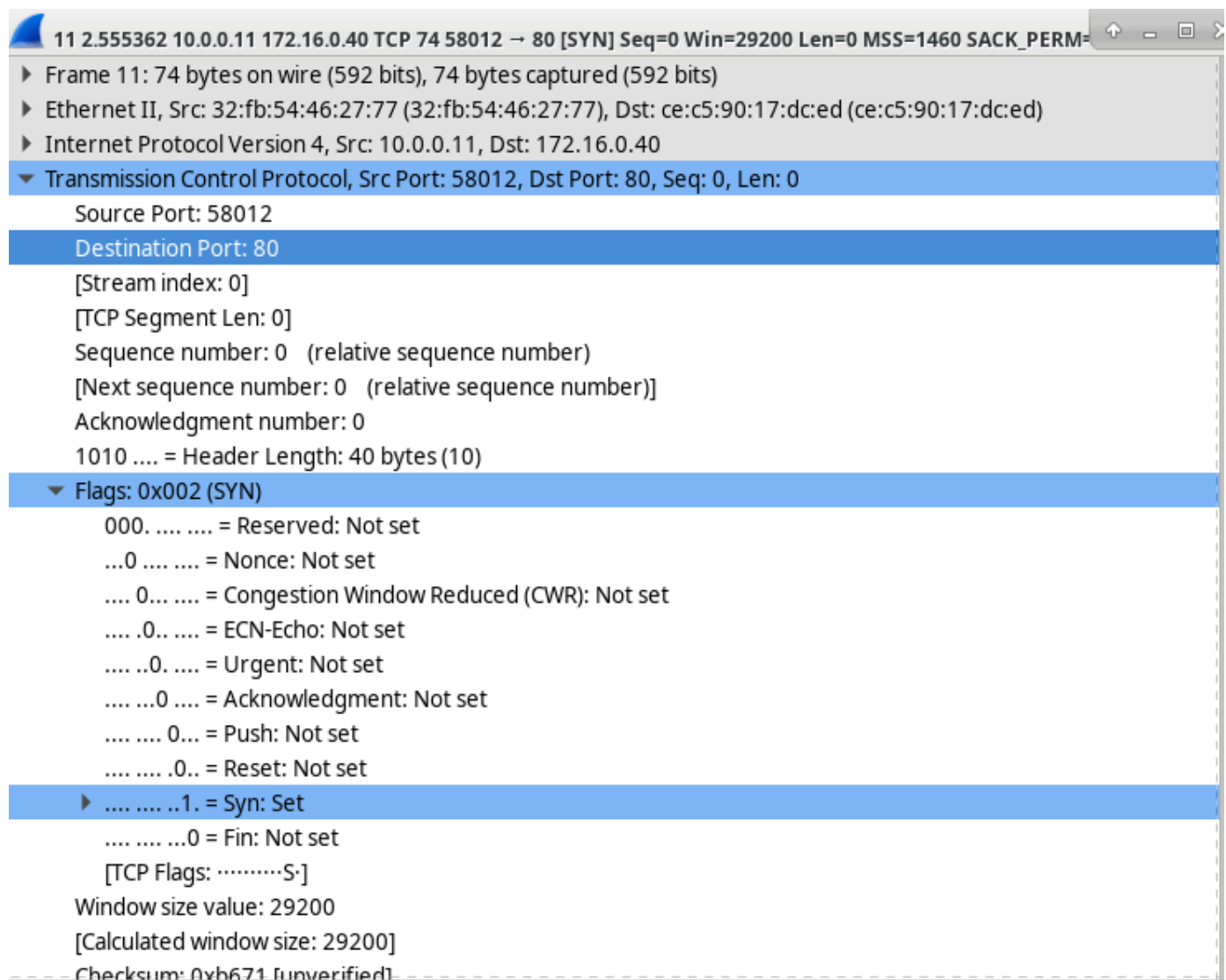
Filter: tcp						
No.	Time	Source	Destination	Protocol	Length	Info
11	2.555362	10.0.0.11	172.16.0.40	TCP	74	58012 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=243037328
12	2.555407	172.16.0.40	10.0.0.11	TCP	74	80 → 58012 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=
13	2.555414	10.0.0.11	172.16.0.40	TCP	66	58012 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2430373282 TSecr=33353534
14	2.555508	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1
15	2.555516	172.16.0.40	10.0.0.11	TCP	66	80 → 58012 [ACK] Seq=1 Ack=293 Win=30208 Len=0 TSval=3335353413 TSecr=243037
16	2.556180	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
17	2.556184	10.0.0.11	172.16.0.40	TCP	66	58012 → 80 [ACK] Seq=293 Ack=325 Win=30720 Len=0 TSval=2430373283 TSecr=3335
46	12.606804	10.0.0.11	172.16.0.40	TCP	66	[TCP Keep-Alive] 58012 → 80 [ACK] Seq=292 Ack=325 Win=30720 Len=0 TSval=2430383
47	12.607159	172.16.0.40	10.0.0.11	TCP	66	[TCP Keep-Alive ACK] 80 → 58012 [ACK] Seq=325 Ack=293 Win=30208 Len=0 TSval=3335

## STEP 2

- Il frame 1 è l'inizio del "three-way handshake" tra il PC e il server su H4. Seleziono il primo pacchetto nella finestra dei dettagli.
- Clicco sulla freccia a sinistra del TCP nella finestra dei dettagli per espandere le informazioni sul TCP.
- Clicco sulla freccia accanto alle flag.

### Informazioni trovare:

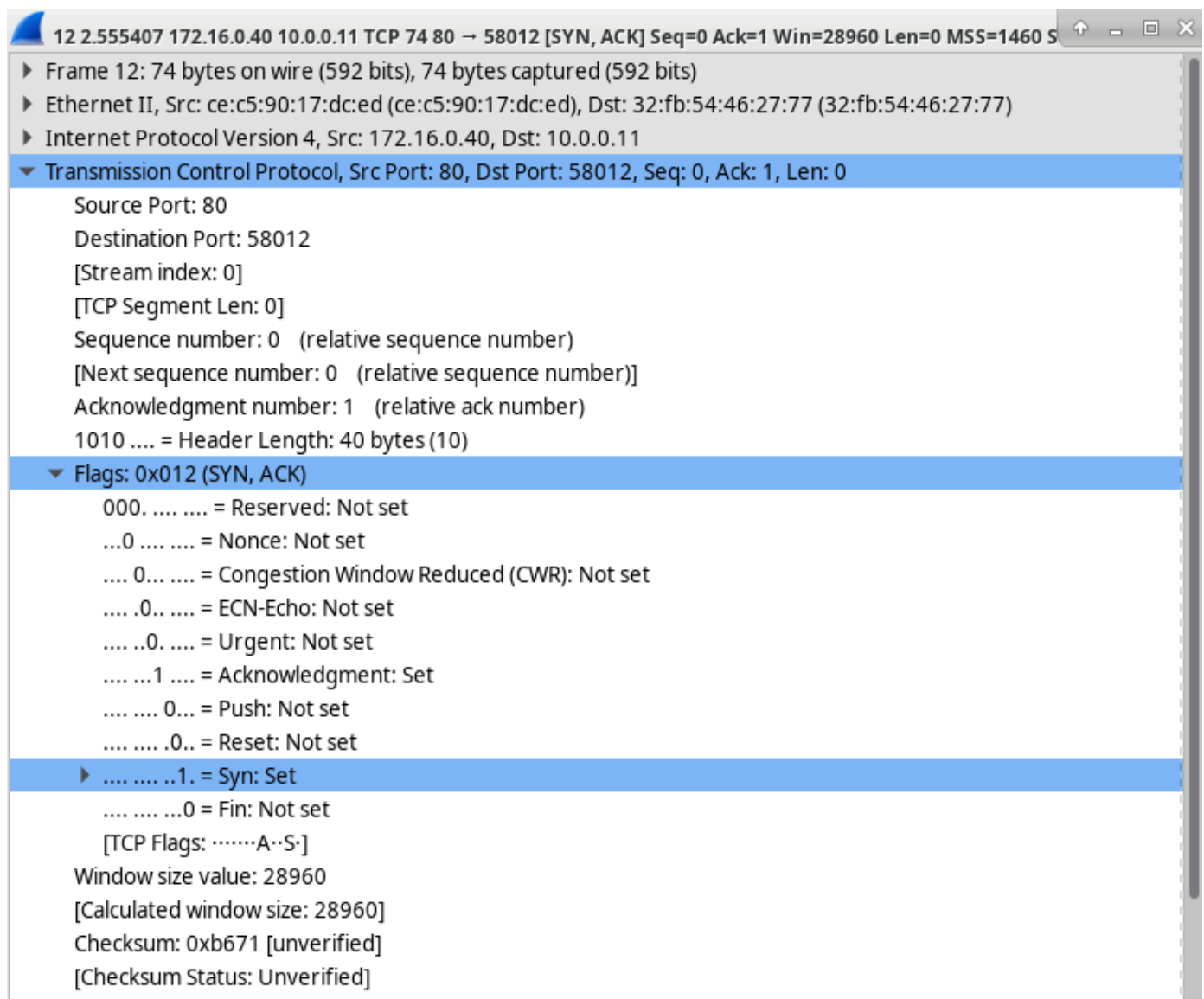
- Porta di origine: **58012**.
- Numero di porta di destinazione TCP: **80**.
- Flag impostata: **SYN**.
- Numero di sequenza relativo: **0**.



- Seleziono il pacchetto successivo nel "three-way handshake". In questo esempio, è il frame 2, che rappresenta la risposta del server web alla richiesta iniziale di avviare una sessione.

#### Informazioni trovare:

- Porta di origine: **80**.
- Porta di destinazione: **58012**.
- Flags: **ACK SYN**.
- Numero di sequenza relativo è **0**.
- Numero di conferma relativo è **1**.

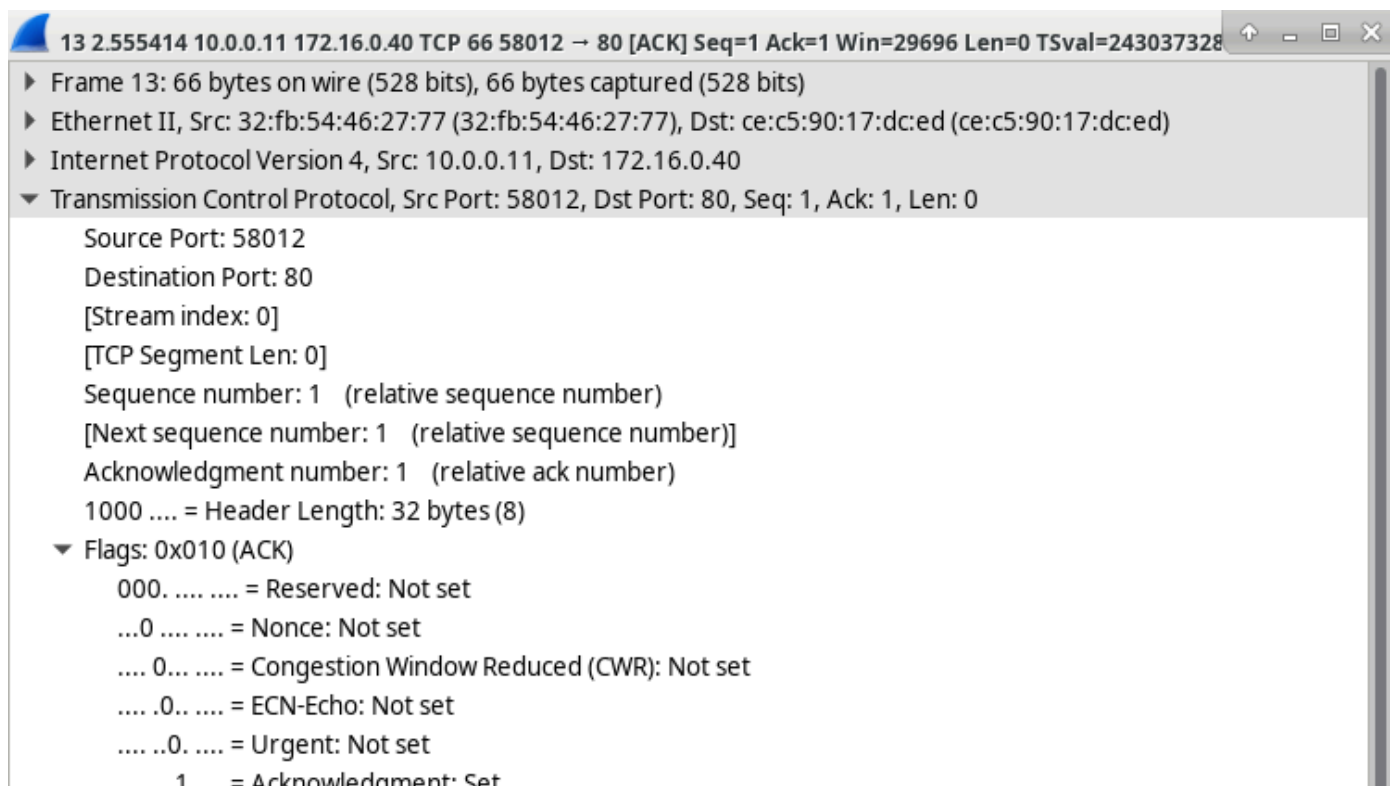


- Infine, seleziona il terzo pacchetto nel "three-way handshake".

### Informazioni trovare:

- Flag: **ACK**.
- Numeri di sequenza e di conferma relativi: **1**.

La connessione TCP è ora stabilita e può iniziare la comunicazione tra il computer sorgente e il server web.



### Parte 3: Visualizzare i Pacchetti con tcpdump

- **Apri una nuova finestra del terminale e digita `man tcpdump`. Nota: potrebbe essere necessario premere ENTER per visualizzare il prompt.**

```
man tcpdump
```

Nelle pagine del manuale di Linux, puoi leggere o cercare le opzioni per selezionare le informazioni desiderate dal file pcap.

- **Nel terminale, apro il file di cattura per visualizzare i primi 3 pacchetti TCP catturati:**

```
tcpdump -r /home/analyst/capture.pcap tcp -c 3
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
09:53:43.955692 IP secOps.58012 > 172.16.0.40.http: Flags [S], seq 3562002547, win 29200, options [mss 1460
,sackOK,TS val 2430373282 ecr 0,nop,wscale 9], length 0
09:53:43.955737 IP 172.16.0.40.http > secOps.58012: Flags [S.], seq 841371516, ack 3562002548, win 28960, o
ptions [mss 1460,sackOK,TS val 3335353413 ecr 2430373282,nop,wscale 9], length 0
09:53:43.955744 IP secOps.58012 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 24303
73282 ecr 3335353413], length 0
```

- **Torno al terminale usato per avviare Mininet. Termino Mininet digitando quit.**

```
mininet> exit
*** Stopping 0 controllers

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



- Dopo aver terminato Mininet, digito `sudo mn -c` per pulire i processi avviati da Mininet.

```
[analyst@sec0ps ~]$ sudo mn -c
[sudo] password for analyst:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old Xii tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_.,:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```