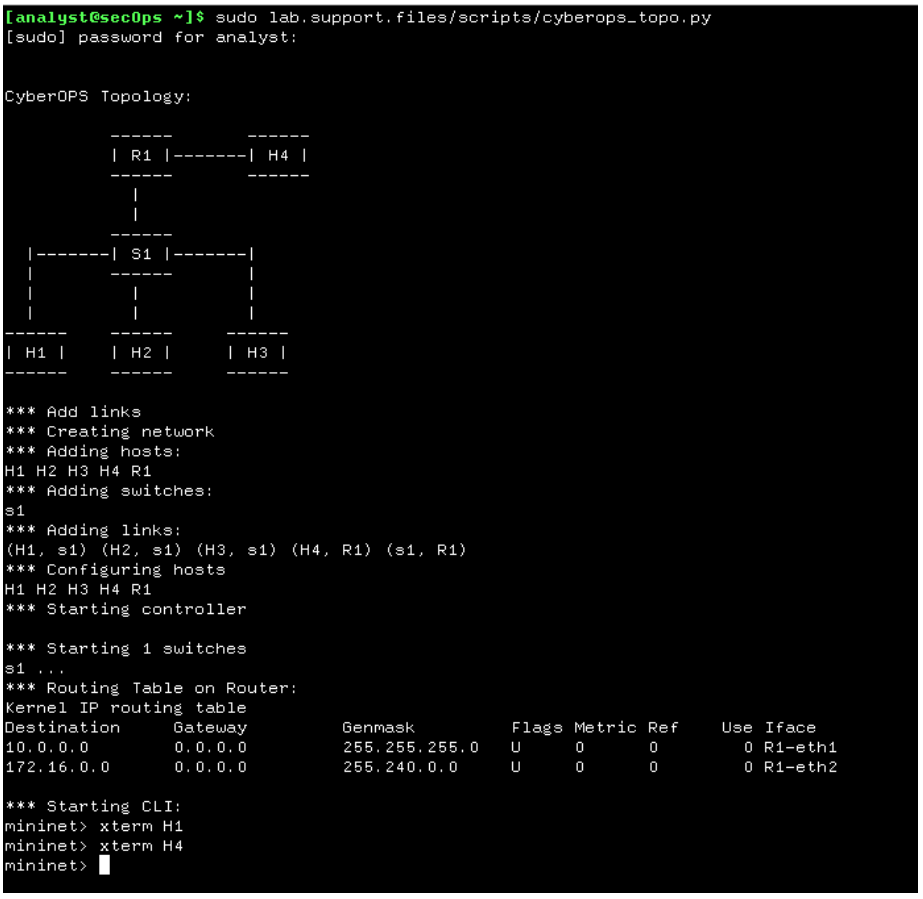
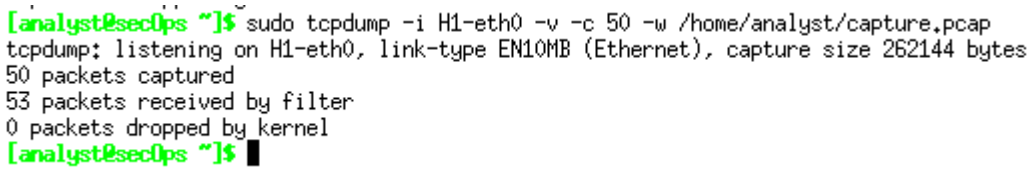
Esercizio L11S2

1. Preparare gli Host per Catturare il Traffico

A. Avviare Mininet dalla VM CyberOps.

- Avviare gli host H1 e H4
- Avviare il webserver da H4
- Avviare il browser web da H1
- Avviare <tcpdump> da H1 creando un file “capture.pcap”
- Navigare su “172.16.0.40” per catturare i pacchetti



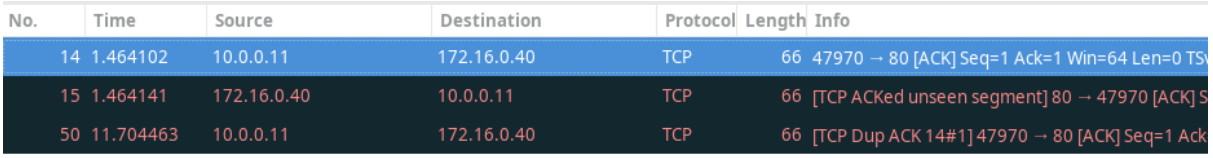


2. Analizzare i pacchetti usando Wireshark

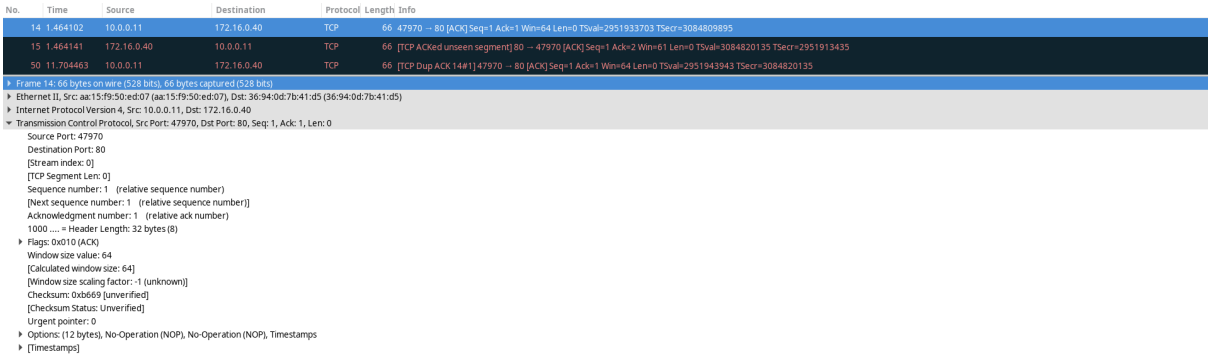
Obiettivo: Applicare un filtro alla cattura salvata.

- **Aprire Wireshark da H1**
- **Aprire il file creato prima**
- **Applicare un filtro “tcp” nella ricerca**
- **Esaminare le info nei pacchetti tcp**

A. Primo pacchetto



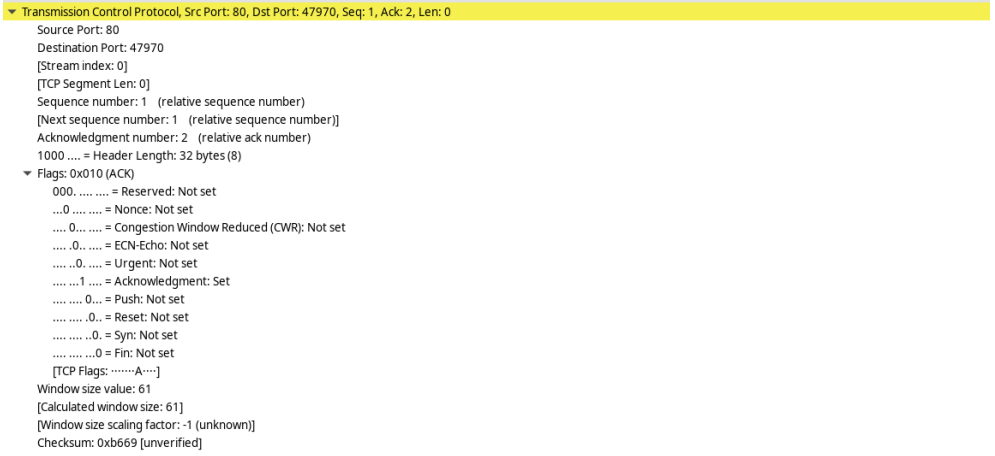
No.	Time	Source	Destination	Protocol	Length	Info
14	1.464102	10.0.0.11	172.16.0.40	TCP	66	47970 → 80 [ACK] Seq=1 Ack=1 Win=64 Len=0 TS
15	1.464141	172.16.0.40	10.0.0.11	TCP	66	[TCP ACKed unseen segment] 80 → 47970 [ACK] S
50	11.704463	10.0.0.11	172.16.0.40	TCP	66	[TCP Dup ACK 14#1] 47970 → 80 [ACK] Seq=1 Ack



```
No.  Time      Source        Destination   Protocol Length Info
  14 1.464102  10.0.0.11     172.16.0.40   TCP      66     47970 → 80 [ACK] Seq=1 Ack=1 Win=64 Len=0 TSval=2951933703 TSecr=3084809895
  15 1.464141  172.16.0.40   10.0.0.11     TCP      66     [TCP ACKed unseen segment] 80 → 47970 [ACK] Seq=1 Ack=2 Win=61 Len=0 TSval=3084820135 TSecr=2951913435
  50 11.704463 10.0.0.11     172.16.0.40   TCP      66     [TCP Dup ACK 14#1] 47970 → 80 [ACK] Seq=1 Ack=1 Win=64 Len=0 TSval=2951943943 TSecr=3084820135
▸ Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: aa:15:f9:50:ed:07 (aa:15:f9:50:ed:07), Dst: 36:94:0d:7b:41:d5 (36:94:0d:7b:41:d5)
▸ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
▾ Transmission Control Protocol, Src Port: 47970, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 47970
    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)
    Window size value: 64
    [Calculated window size: 64]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xb669 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [Timestamps]
```

Qual è il numero di porta TCP di origine? 47970
Come classificheresti la porta di origine? Dinamica/Effimera (porta superiore a 1024 quindi porta non nota)
Qual è il numero di porta TCP di destinazione? 80
Come classificheresti la porta di destinazione? Porta standard utilizzata per il traffico HTTP
Quale flag è impostato? Il flag impostato è ACK
A quale valore è impostato il numero di sequenza relativo? Il numero di sequenza relativo è impostato a 1

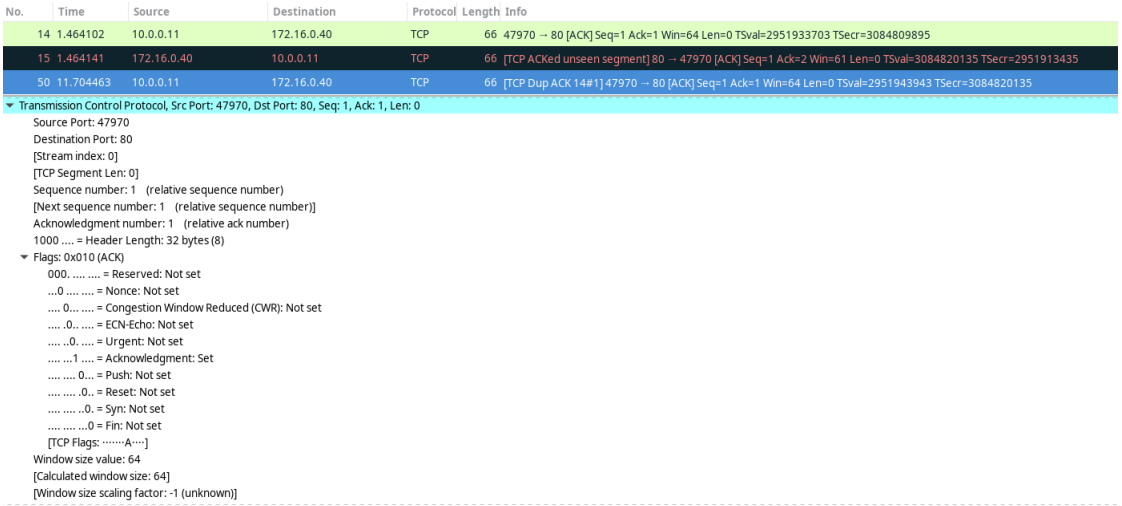
B. Secondo pacchetto



```
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 47970, Seq: 1, Ack: 2, Len: 0
    Source Port: 80
    Destination Port: 47970
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 2    (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····]
    Window size value: 61
    [Calculated window size: 61]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xb669 [unverified]
```

Quali sono i valori delle porte di origine e destinazione? 80
Quali flag sono impostati? Il flag impostato è ACK
A quali valori sono impostati i numeri relativi di sequenza e acknowledgment?
Numero di sequenza relativo è 1, numero di acknowledgment relativo è 2

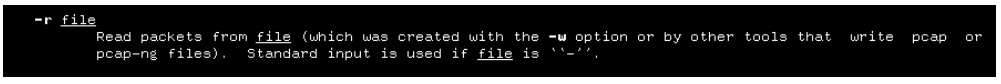
C. Terzo pacchetto



Quale flag è impostato? ACK

3. Visualizzare i pacchetti usando tcpdump

A. Aprire il manuale di tcpdump e trovare opzione “-r”



Cosa fa l'opzione -r? Legge i pacchetti trascritti nei file di estensione .pcap creati con il comando -w

B. Visualizzare i pacchetti tcp catturati in precedenza

```
[analyst@secOps ~]$ man tcpdump
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap -c 3
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
10:32:40.723974 IP 10.0.0.11.57122 > 209-165-200-235.got.net.domain: 22547+ A? www.google.com. (32)
10:32:40.723992 IP 10.0.0.1 > 10.0.0.11: ICMP net 209-165-200-235.got.net unreachable, length 68
10:32:40.723997 IP 10.0.0.11.57122 > 209-165-200-235.got.net.domain: 8989+ AAAA? www.google.com. (32)
[analyst@secOps ~]$
```

C. Chiudere e pulire i processi avviati da Mininet nel terminale

```
*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
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 ~]$
```

```
[analyst@secOps ~]$ 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 udpbwtest mnexec ivs
2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbwtest mnexec i
vs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 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.
[analyst@secOps ~]$
```

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.

- “tcp.port”: utile per visualizzare il traffico sulla porta TCP
- “icmp”: utile per mostrare messaggi d’errore tra i vari dispositivi di rete
- “smb”: utile per rivelare accessi non autorizzati o monitorare il traffico condiviso

In quali altri modi Wireshark potrebbe essere utilizzato in una rete di produzione?

- Security analysis: Rilevamento attacchi e traffico malevolo
- Performance troubleshooting: Analisi latenze e pacchetti persi
- Protocol debugging: Risoluzione problemi applicativi
- Network forensics: Investigazione incidenti di sicurezza