

S11-L3

Obiettivi

Parte 1: Preparare gli Host per Catturare il Traffico

Parte 2: Analizzare i Pacchetti utilizzando Wireshark

Parte 3: Visualizzare i Pacchetti utilizzando tcpdump

Background / Scenario

In questo laboratorio, utilizzerai Wireshark per catturare ed esaminare i pacchetti generati tra il browser del PC utilizzando il Protocollo di Trasferimento Iper-testuale (HTTP) e un server web, come www.google.com. Quando un'applicazione, come HTTP o il Protocollo di Trasferimento File (FTP), inizia per la prima volta su un host, TCP utilizza la stretta di mano a tre vie per stabilire una sessione TCP affidabile tra i due host. Ad esempio, quando un PC utilizza un browser web per navigare su internet, viene avviata una stretta di mano a tre vie, e si stabilisce una sessione tra l'host PC e il server web. Un PC può avere sessioni TCP multiple, simultanee e attive con vari siti web.

Nota per l'Istruttore: L'uso di un analizzatore di pacchetti, come Wireshark, può essere considerato una violazione della politica di sicurezza della scuola. Si raccomanda di ottenere il permesso prima di utilizzare Wireshark per questo laboratorio. Se l'uso di un analizzatore di pacchetti è un problema, l'istruttore potrebbe voler assegnare il laboratorio come compito a casa o eseguire una dimostrazione guidata.

Risorse Richieste

Macchina virtuale CyberOps Workstation

Istruzioni

Ecco la traduzione mantenendo il formato:

Parte 1: Preparare gli Host per Catturare il Traffico

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

b. Avvia Mininet.

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

c. Avvia gli host H1 e H4 in Mininet.

Starting CLI:

```
mininet> xterm H1
```

```
mininet> xterm H4
```

The screenshot displays a Mininet network environment. The main terminal window shows the execution of `sudo lab.support.files/scripts/cyberops_topo.py` and the resulting network topology. The topology diagram shows a central switch `S1` connected to three hosts (`H1`, `H2`, `H3`) and a router (`R1`). The router `R1` is also connected to host `H4`. The terminal output includes the following steps:

- *** 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:

The routing table for the router is shown below:

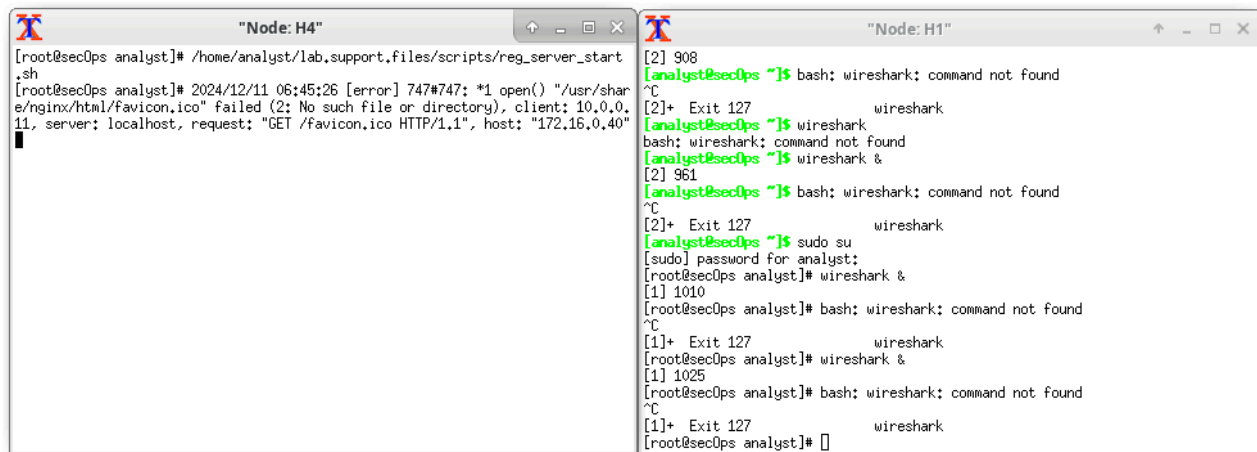
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.0.0.0	0.0.0.0	255.255.255.0	U	0	0		R1-eth1
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0		R1-eth2

The terminal also shows the command `mininet> xterm H1` and `mininet> xterm H4`. Two xterm windows are open, one for host `H4` and one for host `H1`. The `H4` window shows the command `/home/analyst/lab.support.files/scripts/reg_server_start.sh` being executed. The `H1` window shows the command `su analyst` being executed.

d. Avvia il server web su H4.

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

Welcome to nginx!



```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start.sh
[2024/12/11 06:45:26 [error] 747#747: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.0.0.11, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "172.16.0.40"]

[2] 908
[analyst@secOps ~]$ bash; wireshark; command not found
^C
[2]+  Exit 127                  wireshark
[analyst@secOps ~]$ wireshark
bash: wireshark: command not found
[analyst@secOps ~]$ wireshark &
[2] 961
[analyst@secOps ~]$ bash; wireshark; command not found
^C
[2]+  Exit 127                  wireshark
[analyst@secOps ~]$ sudo su
[sudo] password for analyst:
[root@secOps analyst]# wireshark &
[1] 1010
[root@secOps analyst]# bash; wireshark; command not found
^C
[1]+  Exit 127                  wireshark
[root@secOps analyst]# wireshark &
[1] 1025
[root@secOps analyst]# bash; wireshark; command not found
^C
[1]+  Exit 127                  wireshark
[root@secOps analyst]#
```

e. Per motivi di sicurezza, non è possibile eseguire Firefox dall'account utente root. Su H1, usa il comando **switch user** per passare dall'utente root all'utente analyst:

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

f. Avvia il browser web su H1. Questo richiederà alcuni momenti.

```
[analyst@secOps ~]$ firefox &
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

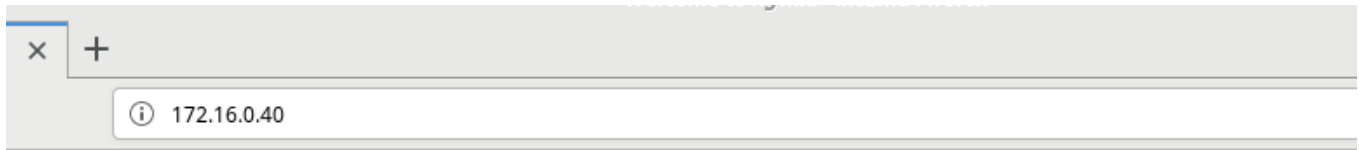
Thank you for using nginx.

g. Dopo che la finestra di Firefox si apre, avvia una sessione **tcpdump** nel terminale del nodo H1 e invia l'output a un file chiamato **capture.pcap**. Con l'opzione **-v**, puoi osservare i progressi. Questa cattura si fermerà automaticamente dopo aver catturato 50 pacchetti, grazie all'opzione **-c 50**.

```
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
```

...

h. Dopo aver avviato **tcpdump**, naviga rapidamente a **172.16.0.40** nel browser web Firefox.



Welcome to nginx!

Parte 2: Analizzare i Pacchetti utilizzando Wireshark

Passaggio 1: Applicare un filtro alla cattura salvata.

a. Premi **INVIO** per visualizzare il prompt. Avvia Wireshark sul nodo H1. Fai clic su **OK** quando viene visualizzato l'avviso relativo all'esecuzione di Wireshark come superutente.

```
[analyst@secOps ~]$ wireshark &
```

b. In Wireshark, fai clic su **File > Open**. Seleziona il file pcap salvato situato in **/home/analyst/capture.pcap**.

c. Applica un filtro **tcp** alla cattura. In questo esempio, i primi 3 frame rappresentano il traffico di interesse.

capture.pcap [Wireshark 2.5.1]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: tcp Expression... Clear Apply Save

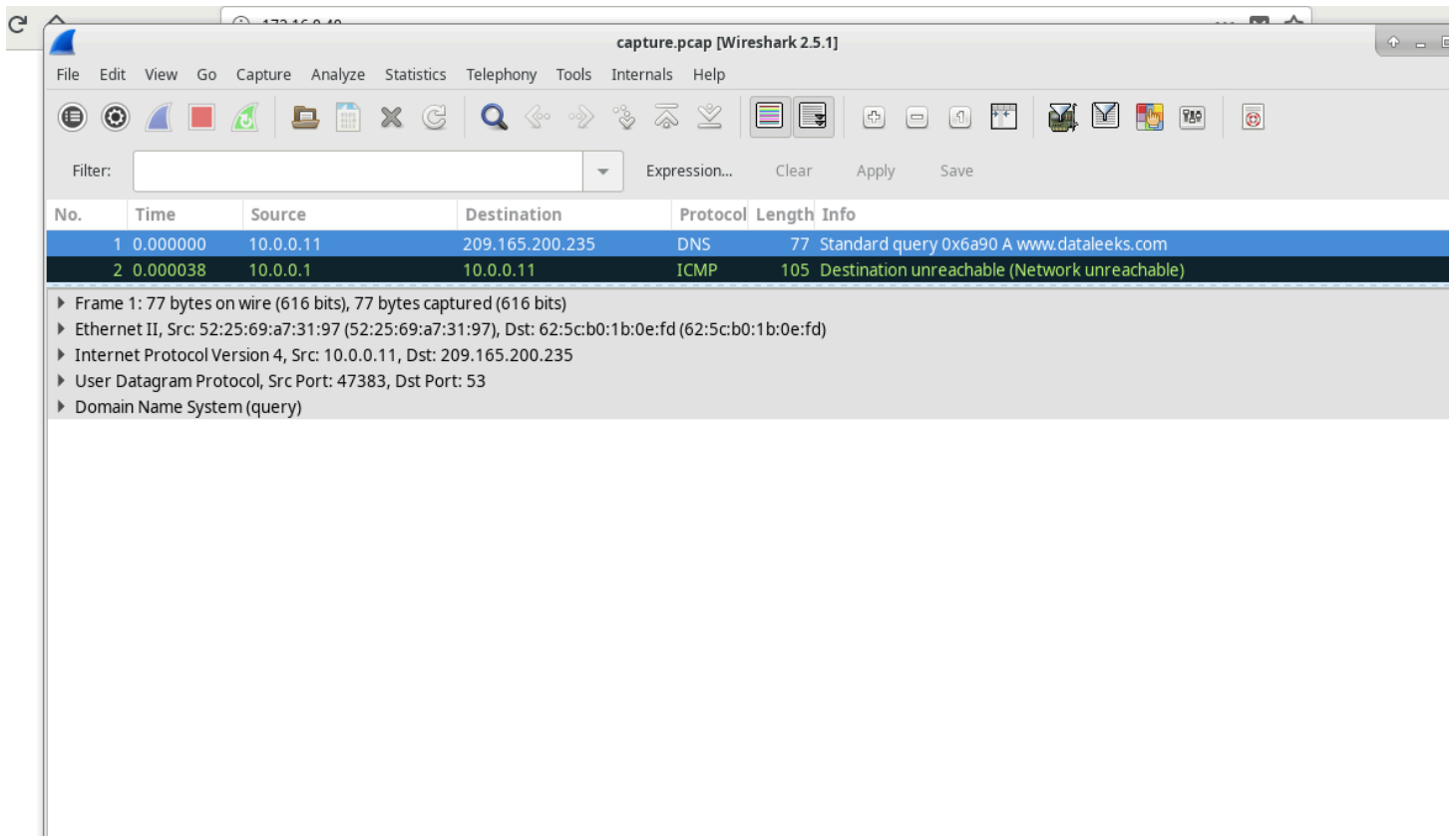
No.	Time	Source	Destination	Protocol	Length	Info
30	22.712414	10.0.0.11	172.16.0.40	TCP	74	60014 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=336621266
31	22.712461	172.16.0.40	10.0.0.11	TCP	74	80 → 60014 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=
32	22.712470	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=3366212666 TSer=319376691
33	22.712590	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
34	22.712599	172.16.0.40	10.0.0.11	TCP	66	80 → 60014 [ACK] Seq=1 Ack=312 Win=30208 Len=0 TSval=3193766904 TSer=336621
35	22.715031	172.16.0.40	10.0.0.11	TCP	304	80 → 60014 [PSH, ACK] Seq=1 Ack=312 Win=30208 Len=238 TSval=3193766907 TSer=
36	22.715034	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=312 Ack=239 Win=30720 Len=0 TSval=3366212669 TSer=3193
37	22.715568	172.16.0.40	10.0.0.11	HTTP	678	HTTP/1.1 200 OK (text/html)
38	22.715570	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=312 Ack=851 Win=31744 Len=0 TSval=3366212669 TSer=3193
43	22.964359	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1
44	22.964444	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
45	22.964538	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=604 Ack=1175 Win=32768 Len=0 TSval=3366212918 TSer=319
69	33.031943	10.0.0.11	172.16.0.40	TCP	66	ITCP Keep-Alive 60014 → 80 [ACK] Seq=603 Ack=1175 Win=32768 Len=0 TSval=336622
70	33.031999	172.16.0.40	10.0.0.11	TCP	66	ITCP Keep-Alive ACK 80 → 60014 [ACK] Seq=1175 Ack=604 Win=31232 Len=0 TSval=314
93	43.271912	10.0.0.11	172.16.0.40	TCP	66	ITCP Keep-Alive 60014 → 80 [ACK] Seq=603 Ack=1175 Win=32768 Len=0 TSval=336623
94	43.271984	172.16.0.40	10.0.0.11	TCP	66	ITCP Keep-Alive ACK 80 → 60014 [ACK] Seq=1175 Ack=604 Win=31232 Len=0 TSval=314

▶ Frame 30: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 ▶ Ethernet II, Src: 52:25:69:a7:31:97 (52:25:69:a7:31:97), Dst: 62:5c:b0:1b:0e:fd (62:5c:b0:1b:0e:fd)
 ▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
 ▶ Transmission Control Protocol, Src Port: 60014, Dst Port: 80, Seq: 0, Len: 0

0000 62 5c b0 1b 0e fd 52 25 69 a7 31 97 08 00 45 00 b...R% i.1...E.

Passaggio 2: Esaminare le informazioni all'interno dei pacchetti, inclusi indirizzi IP, numeri di porta TCP e flag di controllo TCP.

a. In questo esempio, il ****frame 1**** rappresenta l'inizio della stretta di mano a tre vie tra il PC e il server su H4. Nel riquadro dell'elenco dei pacchetti (sezione superiore della finestra principale), seleziona il primo pacchetto, se necessario.



b. Clicca sulla freccia a sinistra del **Transmission Control Protocol** nel riquadro dei dettagli del pacchetto per espanderlo ed esaminare le informazioni TCP. Individua le informazioni sulla porta sorgente e di destinazione.

c. Clicca sulla freccia a sinistra dei **Flags**. Un valore pari a **1** indica che il flag è impostato. Individua il flag impostato in questo pacchetto.

****Nota**:** Potrebbe essere necessario regolare le dimensioni delle finestre superiore e centrale in Wireshark per visualizzare le informazioni necessarie.

Qual è il numero di porta sorgente TCP?

Le risposte possono variare. In questo esempio, la porta sorgente è **58716**.

No.	Time	Source	Destination	Protocol	Length	Info
30	22.712414	10.0.0.11	172.16.0.40	TCP	74	60014 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3366212666 TSecr=0 WS=512
31	22.712461	172.16.0.40	10.0.0.11	TCP	74	80 → 60014 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3193766904 TSecr=3366212666
32	22.712470	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=3366212666 TSecr=3193766904
33	22.712590	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
34	22.712599	172.16.0.40	10.0.0.11	TCP	66	80 → 60014 [ACK] Seq=1 Ack=312 Win=30208 Len=0 TSval=3193766904 TSecr=3366212666
35	22.715031	172.16.0.40	10.0.0.11	TCP	304	80 → 60014 [PSH, ACK] Seq=1 Ack=312 Win=30208 Len=238 TSval=3193766907 TSecr=3366212666 (TCP segment c
36	22.715034	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=312 Ack=239 Win=30720 Len=0 TSval=3366212669 TSecr=3193766907
37	22.715568	172.16.0.40	10.0.0.11	HTTP	678	HTTP/1.1 200 OK (text/html)
38	22.715570	10.0.0.11	172.16.0.40	TCP	66	60014 → 80 [ACK] Seq=312 Ack=851 Win=31744 Len=0 TSval=3366212669 TSecr=3193766907
43	22.964359	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1

▼ Transmission Control Protocol, Src Port: 60014, Dst Port: 80, Seq: 0, Len: 0
Source Port: 60014
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]

Come classificherei la porta sorgente?

Dinamica o **Privata**

Qual è il numero di porta di destinazione TCP?

Porta 80

Come classificherei la porta di destinazione?

Ben nota, registrata (**HTTP** o protocollo web)

Quale flag (o quali flag) è impostato?

Flag SYN

A quale valore è impostato il numero di sequenza relativo?

0

d. Seleziona il pacchetto successivo nella stretta di mano a tre vie.

In questo esempio, si tratta del **frame 2**. Questo è il server web che risponde alla richiesta iniziale per avviare una sessione.

Quali sono i valori delle porte sorgente e di destinazione?

La porta sorgente è ora **80**, e la porta di destinazione è ora **58716**.

Quali flag sono impostati?

Il **flag di riconoscimento (ACK)** e il **flag SYN**.

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

Il numero relativo di sequenza è **0**, e il numero relativo di riconoscimento è **1**.

e. Infine, seleziona il terzo pacchetto della stretta di mano a tre vie.

Esamina il terzo e ultimo pacchetto della stretta di mano.

Quale flag (o quali flag) è impostato?

Flag di riconoscimento (ACK)

I numeri relativi di sequenza e riconoscimento sono impostati a **1** come punto di partenza. La connessione TCP è stabilita e la comunicazione tra il computer sorgente e il server web può iniziare.

Ecco la traduzione del testo:

Parte 3: Visualizzare i pacchetti utilizzando tcpdump

Puoi anche visualizzare il file pcap e filtrare per ottenere le informazioni desiderate.

a. Aprire una nuova finestra del terminale e digitare **man tcpdump**.

Nota: Potrebbe essere necessario premere **INVIO** per visualizzare il prompt.

Usando le pagine del manuale disponibili nel sistema operativo Linux, puoi leggere o cercare attraverso le opzioni per selezionare le informazioni desiderate dal file pcap.

```
[analyst@secOps ~]$ man tcpdump
```


****TCPDUMP(1)****

Manuale dei comandi generali ****TCPDUMP(1)****

****NOME****

tcpdump - cattura il traffico su una rete

****SINOSSI****

```
tcpdump [ -AbdDefhHIJKILnNOpqStuUvxX# ] [ -B buffer_size ]  
[ -c count ]  
[ -C file_size ] [ -G rotate_seconds ] [ -F file ]  
[ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]  
[ --number ] [ -Q in|out|inout ]  
[ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]  
[ -W filecount ]  
[ -E spi@ipaddr algo:secret,... ]  
[ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]  
[ --time-stamp-precision=tstamp_precision ]  
[ --immediate-mode ] [ --version ]  
[ expression ]
```

Per cercare nelle pagine del manuale, puoi usare ****/**** (cerca in avanti) o ****?**** (cerca all'indietro) per trovare termini specifici, e ****n**** per passare al prossimo risultato o ****q**** per uscire.

Ad esempio, cerca informazioni sull'opzione ****-r****, digitando ****/-r****. Digita ****n**** per spostarti al prossimo risultato.

Cosa fa l'opzione **-r?**

L'opzione ****-r**** consente di leggere i pacchetti da un file salvato utilizzando l'opzione ****-w**** di tcpdump o altri strumenti che generano file pcap o pcap-ng, come Wireshark.

b. Nello stesso terminale, apri il file di cattura utilizzando il seguente comando per visualizzare i primi 3 pacchetti TCP catturati:

```
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
```

****Output**:**

reading from file capture.pcap, link-type EN10MB (Ethernet)

13:58:30.647462 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [S], seq 2432755549, win 29200, options [mss 1460,sackOK,TS val 3864513189 ecr 0,nop,wscale 9], length 0

13:58:30.647543 IP 172.16.0.40.http > 10.0.0.11.58716: Flags [S.], seq 1766419191, ack 2432755550, win 28960, options [mss 1460,sackOK,TS val 50557410 ecr 3864513189,nop,wscale 9], length 0

13:58:30.647544 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [.), ack 1, win 58, options [nop,nop,TS val 3864513189 ecr 50557410], length 0

Per visualizzare la stretta di mano a tre vie, potrebbe essere necessario aumentare il numero di linee dopo l'opzione ****-c****.

c. Navigare nel terminale usato per avviare Mininet. Termina Mininet digitando ****quit**** nella finestra principale del terminale CyberOps VM.

```
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 ~]$
```

d. Dopo aver terminato Mininet, inserisci ****sudo mn -c**** per pulire i processi avviati da Mininet. Inserisci la password ****cyberops**** quando richiesto.

```
[analyst@secOps ~]$ sudo mn -c
```

[sudo] password for analyst: