



S1-S-L3

Utilizzo di Wireshark per Osservare la Stretta
di Mano TCP a 3 Vie

TRACCIA

In questo laboratorio, completeremo 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



PREPARAZIONE HOST PER CATTURA

Come prima cosa apriamo il nostro terminale e inseriamo il comando “sudo lab.support.files/scripts/cyberops_topo.py”. Questo script viene utilizzato all'interno della VM di cyberops per configurare una topologia di rete simulata. Dopodichè creiamo i nostri host attraverso il comando “xterm H1” per l'host H1 e “xterm H4” per l'host H4.

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

CyberOPS Topology:
  -----
  | R1 | -----| H4 |
  -----
  |
  |
  -----
  | S1 | -----|
  |
  |
  -----
  | H1 | -----| H2 | -----| H3 |
  |

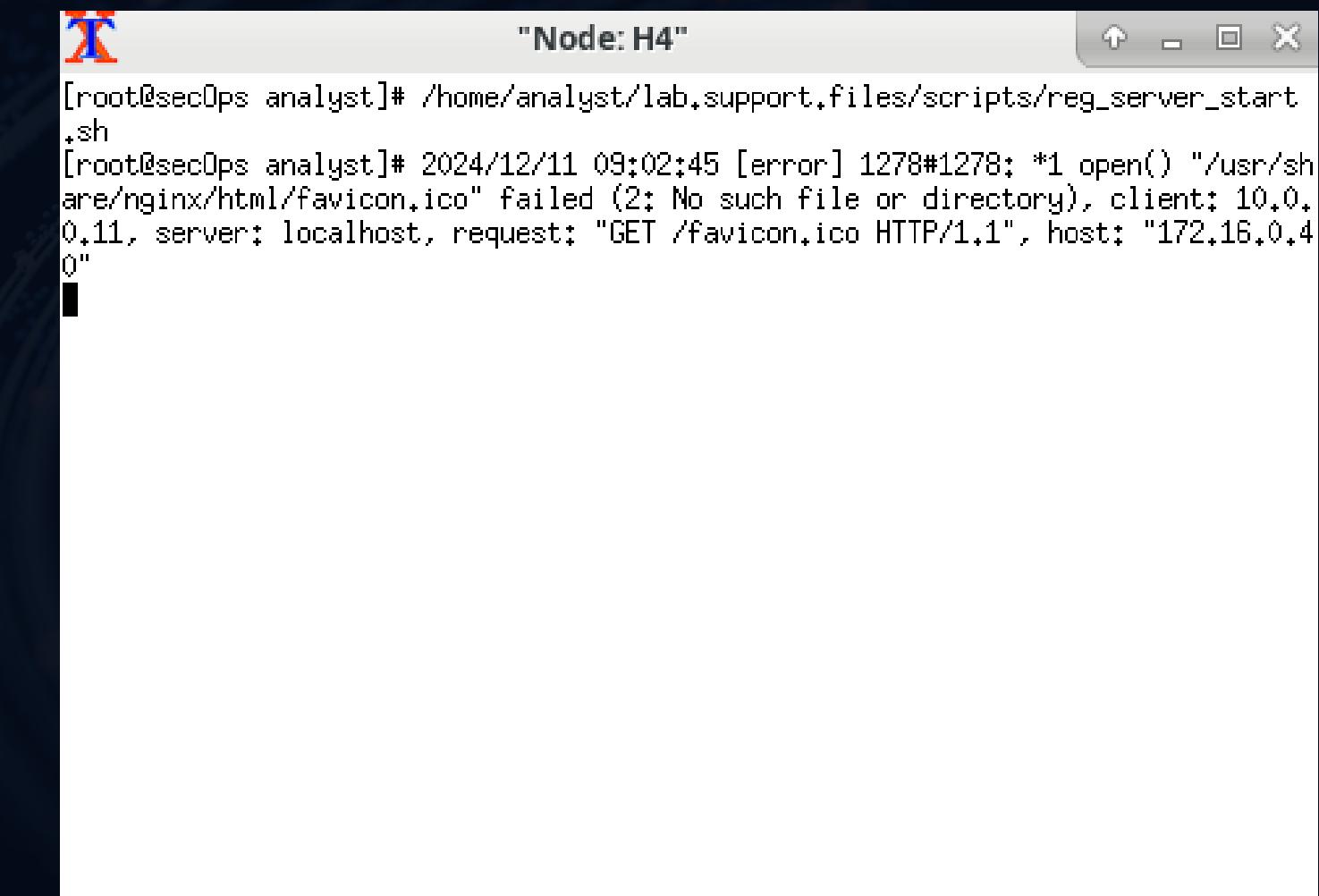
*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
(H1, s1) (H2, s1) (H3, s1) (H4, R1) (s1, R1)
*** Configuring hosts
H1 H2 H3 H4 R1
*** Starting controller

*** 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> xterm H1
mininet> xterm H4
mininet> █
```

AVVIO SERVER H4

Procediamo poi con l'avvio di un server web che ci servirà al fine di registrare eventi di rete con wireshark

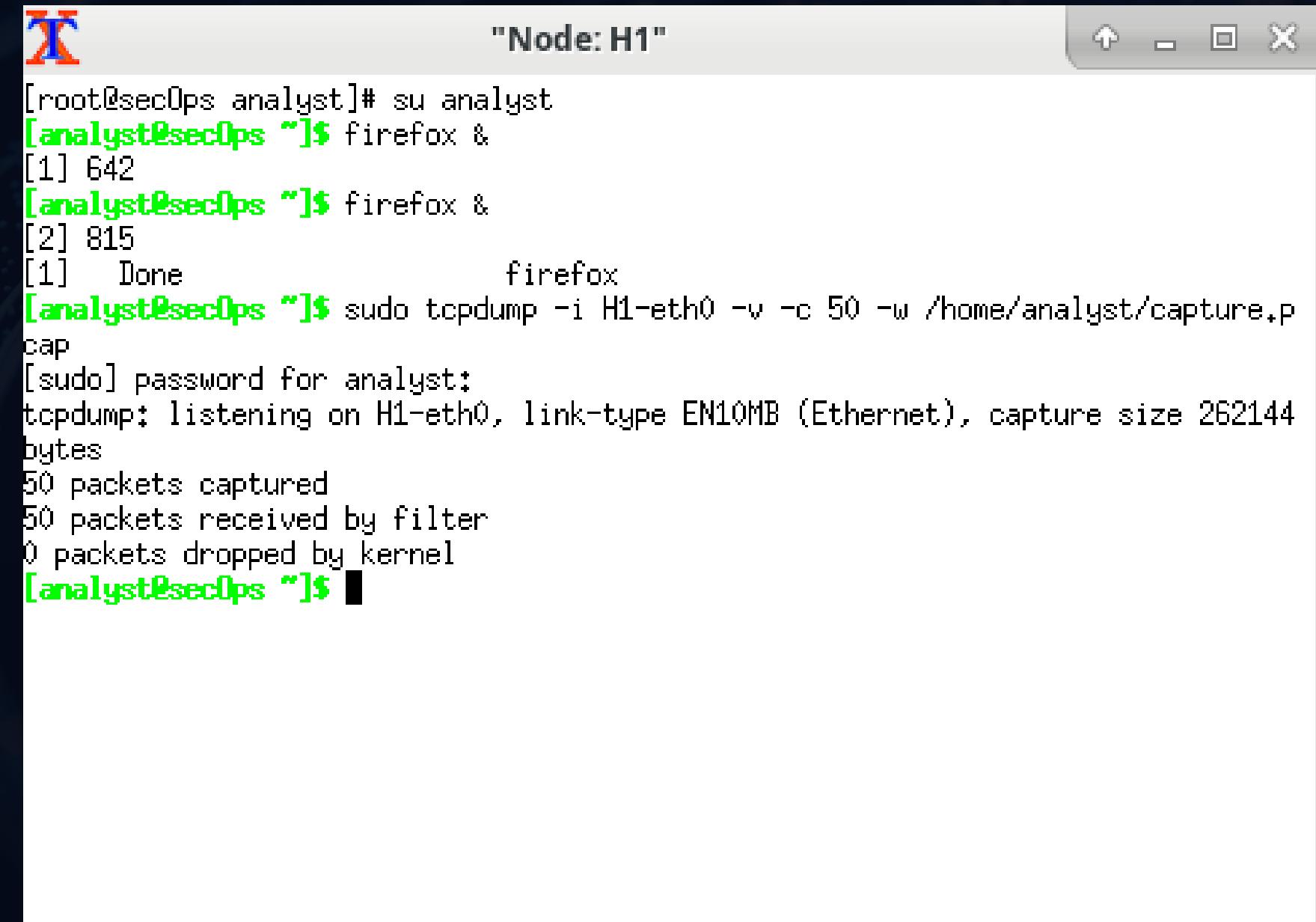


The screenshot shows a terminal window titled "Node: H4". The terminal is running on a Linux system with a root shell. The user has run the command "/home/analyst/lab.support.files/scripts/reg_server_start.sh". The output shows an error message from Nginx: "2024/12/11 09:02:45 [error] 1278#1278: *1 open() \"/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\"".

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start.sh
[root@secOps analyst]# 2024/12/11 09:02:45 [error] 1278#1278: *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\"
```

CATTURA PACCHETTI TCPDUMP

Avviamo poi il nostro browser web firefox attraverso l'apposito comando e mettiamoci a catturare i pacchetti. Ne verranno catturati 50 in quanto nel comando abbiamo stabilito un massimo di 50 pacchetti. Il pacchetto che verrà salvato sarà .pcap il quale è un formato leggibile da strumenti come wireshark e tcpdump



```
[root@secOps analyst]# su analyst
[analyst@secOps ~]$ firefox &
[1] 642
[analyst@secOps ~]$ firefox &
[2] 815
[1] Done firefox
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
[sudo] password for analyst:
tcpdump: listening on H1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
50 packets captured
50 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```

HANDSHAKE 3 WAYS SU WIRESHARK

The screenshot displays the Wireshark interface with the following details:

- File Menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Print, as well as search and analysis tools.
- Filter Bar:** Set to "tcp".
- Panels:**
 - Packet List:** Shows 26 total packets. The first six packets (Frame 20 to Frame 25) are highlighted in blue, indicating they are part of a selected conversation. The last two packets (Frame 26 and 27) are highlighted in green, indicating they are part of another selected conversation.
 - Details:** Shows the raw hex and ASCII data for the selected packet (Frame 20).
 - Bytes:** Shows the raw hex and ASCII data for the selected packet (Frame 20).
- Bottom Status Bar:** Displays the status message: "Frame 20: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)" and the selected source and destination MAC addresses.

No.	Time	Source	Destination	Protocol	Length	Info
20	6.311084	10.0.0.11	172.16.0.40	TCP	74	40530 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3706050147 TSecr=0 WS=512
21	6.311131	172.16.0.40	10.0.0.11	TCP	74	80 → 40530 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3739299551 TSecr=3706050147 WS=512
22	6.311140	10.0.0.11	172.16.0.40	TCP	66	40530 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=3706050147 TSecr=3739299551
23	6.311275	10.0.0.11	172.16.0.40	HTTP	358	GET /favicon.ico HTTP/1.1
24	6.311284	172.16.0.40	10.0.0.11	TCP	66	80 → 40530 [ACK] Seq=1 Ack=293 Win=30208 Len=0 TSval=3739299551 TSecr=3706050147
25	6.311423	172.16.0.40	10.0.0.11	HTTP	390	HTTP/1.1 404 Not Found (text/html)
26	6.311553	10.0.0.11	172.16.0.40	TCP	66	40530 → 80 [ACK] Seq=293 Ack=325 Win=30720 Len=0 TSval=3706050147 TSecr=3739299551

Frame 20: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: fa:f3:73:c6:95:e8 (fa:f3:73:c6:95:e8), Dst: fe:38:6e:2a:55:ea (fe:38:6e:2a:55:ea)
Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
Transmission Control Protocol, Src Port: 40530, Dst Port: 80, Seq: 0, Len: 0

```
0000 fe 38 6e 2a 55 ea fa f3 73 c6 95 e8 08 00 45 00 .8n*U... s.....E.  
0010 00 3c 21 59 40 00 40 06 63 20 0a 00 00 0b ac 10 .<!Y@.@. c ..  
0020 00 28 9e 52 00 50 66 2c d7 a3 00 00 00 00 a0 02 .(.R.Pf, .....  
0030 72 10 b6 71 00 00 02 04 05 b4 04 02 08 0a dc e5 r..q.....  
0040 d6 63 00 00 00 00 01 03 03 09 .c.....
```

HANDSHAKE 3 WAYS SU WIRESHARK

Come si può vedere dall'immagine precedente, abbiamo catturato l'handshake 3ways tra il pc client e il server creato su H4.

- SYN: un pacchetto syn viene inviato dall'IP 10.0.0.11 alla porta 80 dell'indirizzo 172.16.0.40 per iniziare la connessione TCP.
- SYN-ACK: Viene risposto con un syn-ack da 172.16.0.40 che conferma la ricezione del Syn.
- ACK: conferma il completamento dell'handshake da parte del client 10.0.0.11

L'handshake TCP è completato con successo, dimostrando che la connessione tra il client e il server è stata stabilita senza problemi.

HANDSHAKE 3 WAYS SU TCPDUMP

Si può, inoltre, verificare la cattura tramite tcpdump specificando il numero di righe che si vuole vedere e i pacchetti TCP. Ci mostrerà le stesse cose di wireshark ma direttamente in CLI.

```
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap top -c 6
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
09:02:45.020319 IP 10.0.0.11.40530 > 172.16.0.40.http: Flags [S], seq 1714214819, win 29200, options [mss 1460,sackOK,TS val 3706050147 ecr 0,nop,wscale
09:02:45.020366 IP 172.16.0.40.http > 10.0.0.11.40530: Flags [S.], seq 1717130966, ack 1714214820, win 28960, options [mss 1460,sackOK,TS val 3739299551
09:02:45.020375 IP 10.0.0.11.40530 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 3706050147 ecr 3739299551], length 0
09:02:45.020510 IP 10.0.0.11.40530 > 172.16.0.40.http: Flags [P.], seq 1:293, ack 1, win 58, options [nop,nop,TS val 3706050147 ecr 3739299551], length
09:02:45.020519 IP 172.16.0.40.http > 10.0.0.11.40530: Flags [.], ack 293, win 59, options [nop,nop,TS val 3739299551 ecr 3706050147], length 0
09:02:45.020658 IP 172.16.0.40.http > 10.0.0.11.40530: Flags [P.], seq 1:325, ack 293, win 59, options [nop,nop,TS val 3739299551 ecr 3706050147], lengt
[analyst@secOps ~]$
```

**THANKS FOR THE
ATTENTION**

MATTIA DI DONATO