Relazione Wireshark TCP 3-Way Handshake Process

Nella seguente immagine avvio la VM CyberOps Workstation

Dal terminale uso il comando di seguito per utilizzare mininet

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

```
[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>
```

Dopo aver eseguito il comando sopra riportato utilizzo i seguenti comandi per aprire due terminali diversi, nei quali eseguirò i comandi riportati di seguito:

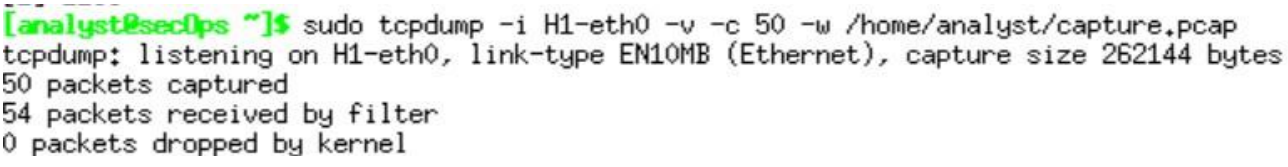
mininet > xterm H1

mininet> xterm H4

Nel terminale **Node: H1** avvio con il comando **firefox &** il browser, che utilizzero per inserire l'IP da scansionare successivamente con il tool **Wireshark**

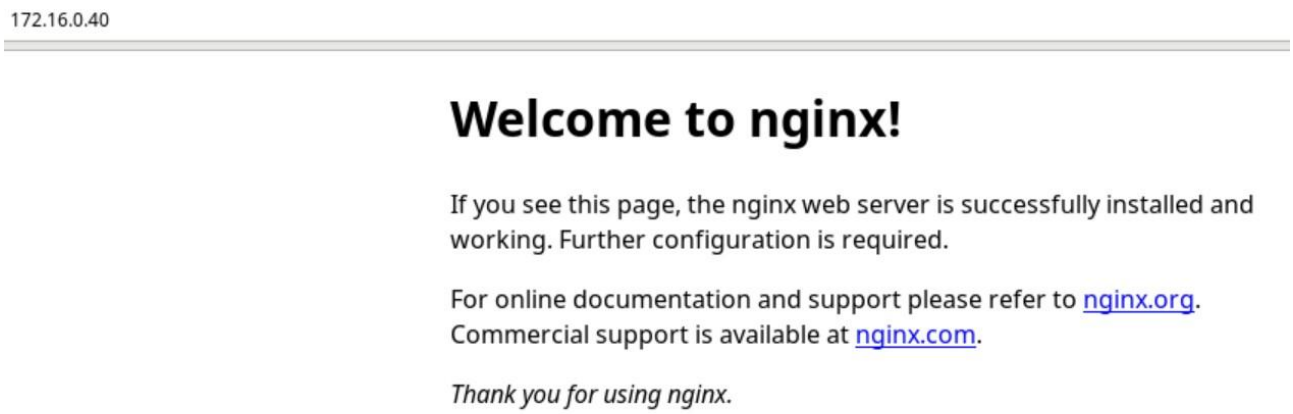
Nel terminale **Node: H4** utilizzo il comando specifico per collegarmi e mettermi in ascolto e con il quale posso vedere i **pacchetti catturati** ed i **pacchetti ricevuti** nella trasmissione:

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

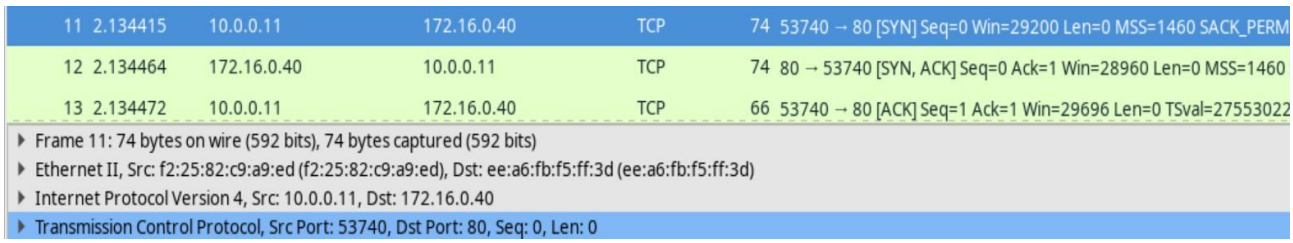


Dopo aver eseguito il comando per mettermi in ascolto, come illustrato in precedenza, vado sulla pagina del Browser e sulla barra URL scrivo l'indirizzo IP sul quale mi metto in ascolto, come di seguito, avrò la pagina **Welcome ti nginx !**

172.16.0.40



Successivamente alle operazioni descritte ed illustrate in precedenza, aprò il tool Wireshark con il quale vedo il traffico di rete, come riportato di seguito



Per la visualizzazione dei processi attivati all'inizio, utilizzo da terminale della **VM CyberOps**

man tcpdump

```
TCPDUMP(1)                  General Commands Manual                  TCPDUMP(1)

NAME
       tcpdump - dump traffic on a network

SYNOPSIS
       tcpdump [ -AbdDefhHIJKlLnNOpqStuUvxX# ] [ -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 operare con **mininet** nel mio terminale utilizzo il seguente comando

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

```
[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>
```

Di seguito per visualizzare i primi 3 pacchetti dati con protocollo TCP, utilizzo il comando scritto di seguito:

[analista@secOps ~]$ tcpdump -r /home/analista/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)
03:51:02.340010 IP 10.0.0.11.53740 > 172.16.0.40.http: Flags [S], seq 179890254,
 win 29200, options [mss 1460,sackOK,TS val 2755302284 ecr 0,nop,wscale 9], leng
th 0
03:51:02.340059 IP 172.16.0.40.http > 10.0.0.11.53740: Flags [S.], seq 419916434
3, ack 179890255, win 28960, options [mss 1460,sackOK,TS val 1665610627 ecr 2755
302284,nop,wscale 9], length 0
03:51:02.340067 IP 10.0.0.11.53740 > 172.16.0.40.http: Flags [.], ack 1, win 58,
 options [nop,nop,TS val 2755302284 ecr 1665610627], length 0
```

In conclusione per chiudere i processi attivi utilizzo il comando **quit**, come illustrato di seguito e per pulire tutti i processi utilizzo il comando **sudo mn -c**

```
*** Starting CLI:
mininet> quit
*** Stopping 0 controllers

*** Stopping 5 links
.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@secOps ~]$ sudo mn -c
*** 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 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 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.
```

Report finale

Dall’analisi finale dei vari processi attivati in precedenza nelle varie illustrazioni, posso identificare che:

- numero della porta sorgente analizzata su Wireshark, da immagini precedenti è la porta:
  - **80**

- la porta sulla quale sono in ascolto (listening) può essere:
  - **privata**
  - **dinamica**

- il numero della porta di destinazione è la porta del protocollo TCP è:
  - **53740**

- il protocollo di destinazione in ascolto è:
  - **http**

- il flag delle 3-Way Handshake è**:**
  - **SYN/ACK**

- Il numero della sequenza di trasmissione è: **0**
- Il numero di conferma di trasmissione è: **1**

23/10/2024 **Christian Vagnoni**