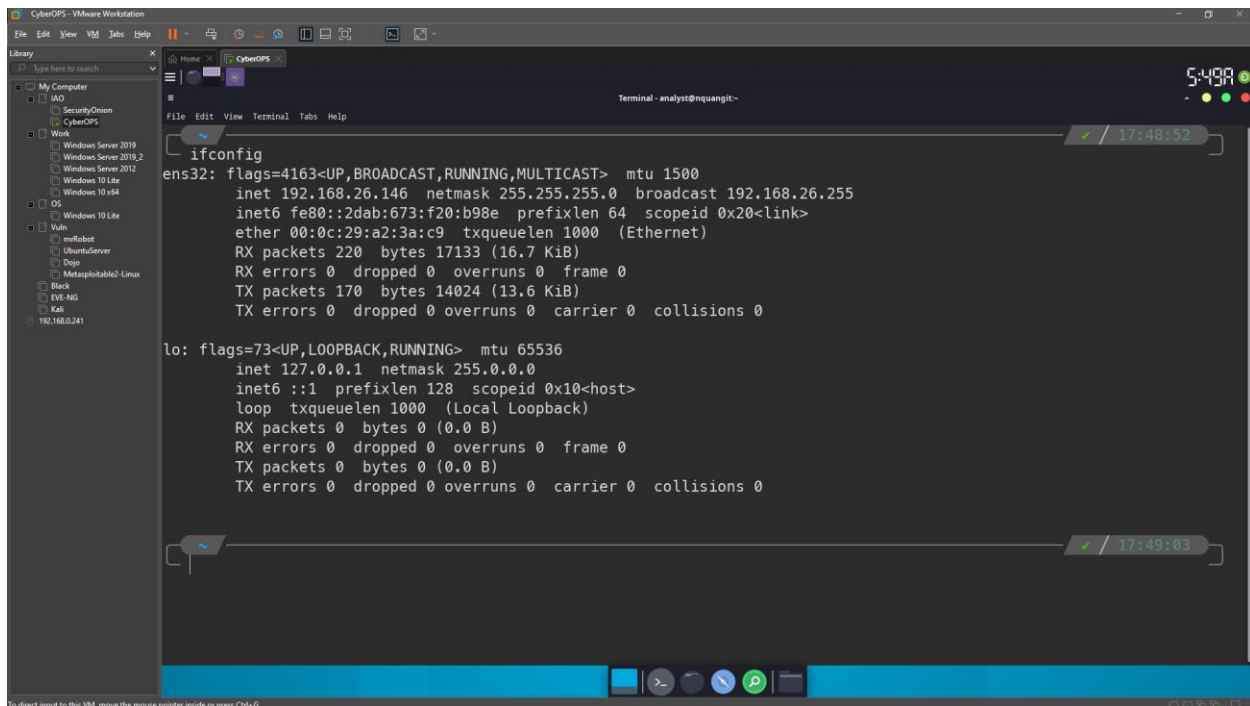


Lab - Introduction to Wireshark

Instructions

Install and Verify the Mininet Topology

Verify your PC's interface addresses.



The screenshot shows a terminal window within a CyberOps virtual machine. The terminal displays the output of the `ifconfig` command for two interfaces: `ens32` and `lo`. The `ens32` interface is an Ethernet card with an IP address of `192.168.26.146` and a netmask of `255.255.255.0`. The `lo` interface is a loopback device with an IP address of `127.0.0.1` and a netmask of `255.0.0.0`. The terminal window has a title bar that reads "CyberOps - VMware Workstation" and a menu bar with "File", "Edit", "View", "Terminal", "Help". The left sidebar shows a library of virtual machines, including "My Computer", "SecurityOption", "CyberOps", "Work", "Windows Server 2019", "Windows Server 2016", "Windows Server 2012", "Windows 10 Lite", "Windows 10 x64", "OS", "Windows 10 Lite", "Vuln", "Metasploit", "UbuntuServer", "Drogo", "Metasploitables2-Linux", "Black", "EVE-NG", "Kali", and "192.168.0.241". The bottom status bar indicates "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

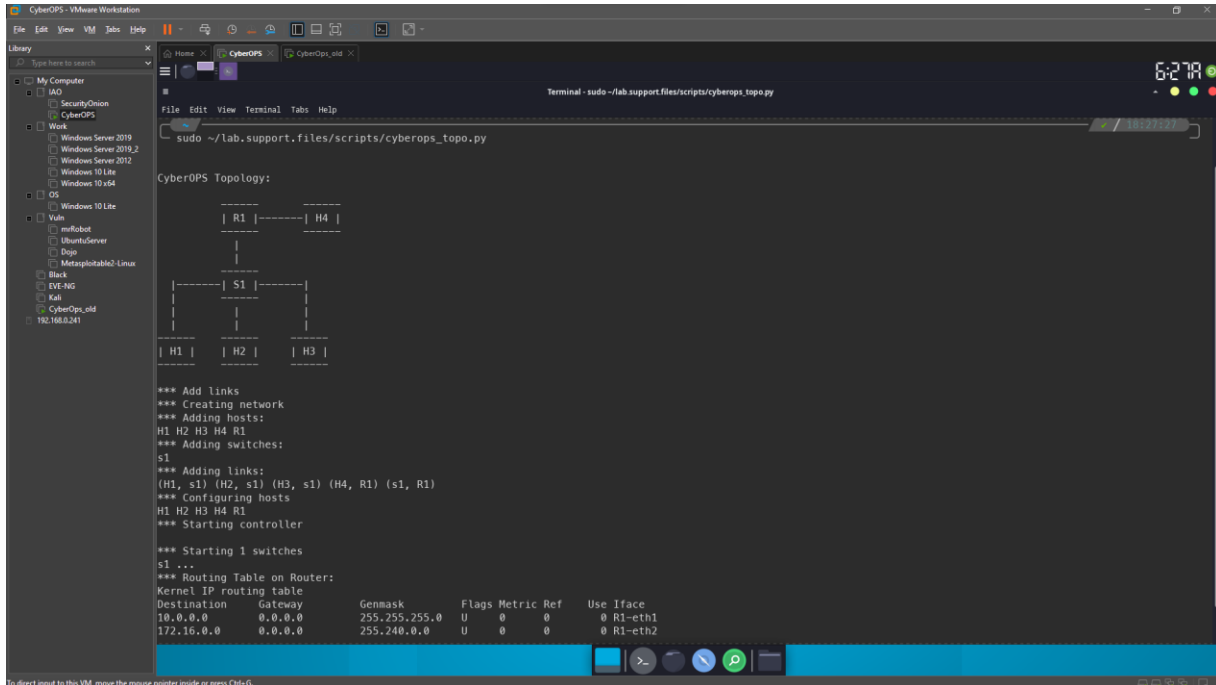
```
CyberOps - VMware Workstation
File Edit View VM Help
Library
Type here to search
My Computer
  IAO
  SecurityOption
  CyberOps
Work
  Windows Server 2019
  Windows Server 2016
  Windows Server 2012
  Windows 10 Lite
  Windows 10 x64
OS
  Windows 10 Lite
Vuln
  Metasploit
  UbuntuServer
  Drogo
  Metasploitables2-Linux
  Black
  EVE-NG
  Kali
  192.168.0.241

Terminal - analyst@nquangit-
File Edit View Terminal Help
17:48:52
ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.26.146 netmask 255.255.255.0 broadcast 192.168.26.255
    inet6 fe80::2dab:673:f20:b98e prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:a2:3a:c9 txqueuelen 1000 (Ethernet)
    RX packets 220 bytes 17133 (16.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 170 bytes 14024 (13.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

17:49:03
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Run the Python script to install the Mininet Topology.



```

CyberOPS - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
  IAO
  Security Onion
  CyberOPS
  Work
    Windows Server 2019
    Windows Server 2019_2
    Windows Server 2012
    Windows 10 Lite
    Windows 10 x64
  OS
    Windows 10 Lite
  Vm
    mRobot
    UbuntuServer
    Dns
    Metasploit62-Linux
    Black
    EVE-NG
    Kali
    CyberOps_old
    192.168.0.241

Terminal - sudo ~/lab.support.files/scripts/cyberops_topo.py
File Edit View Terminal Tabs Help
sudo ~/lab.support.files/scripts/cyberops_topo.py

CyberOPS Topology:

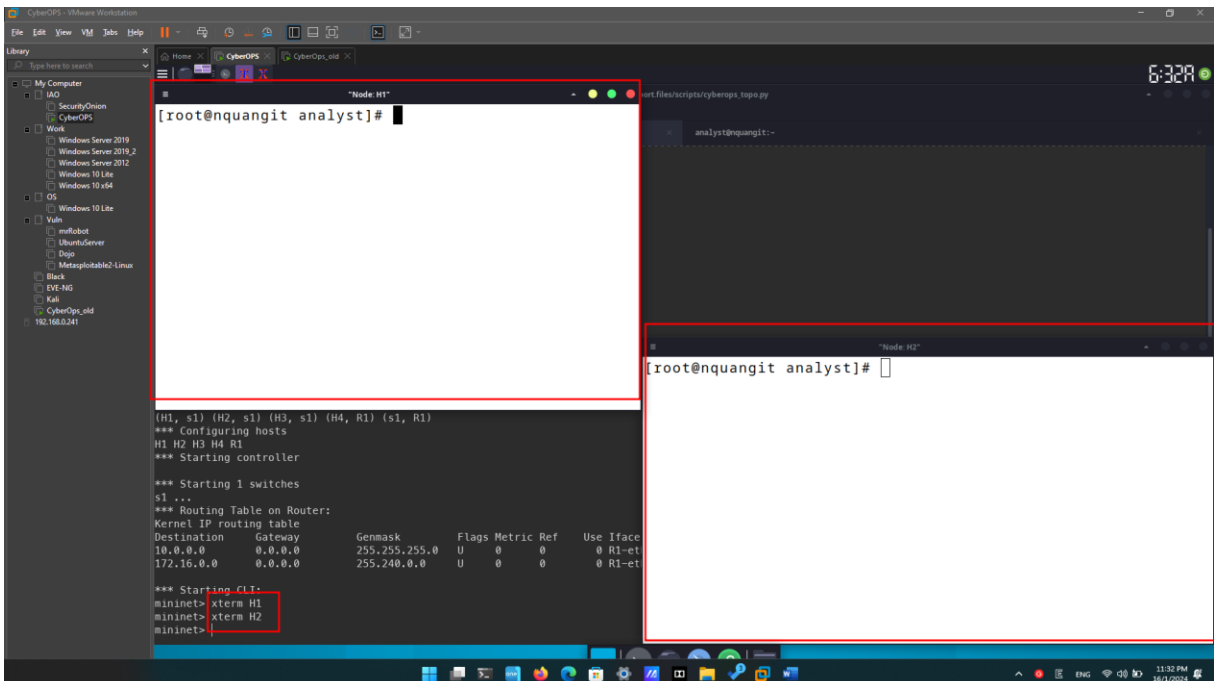
  +-----+
  | R1 |-----| H4 |
  +-----+
  |
  |
  +-----+
  | S1 |-----+
  +-----+
  |
  |
  +-----+
  | H1 | | H2 | | H3 |
  +-----+

*** 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
  
```

Record IP and MAC addresses for H1 and H2.

- At the mininet prompt, start terminal windows on hosts H1 and H2. This will open separate windows for these hosts. Each host will have a separate configuration for the network including unique IP and MAC addresses.



```

CyberOPS - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
  IAO
  Security Onion
  CyberOPS
  Work
    Windows Server 2019
    Windows Server 2019_2
    Windows Server 2012
    Windows 10 Lite
    Windows 10 x64
  OS
    Windows 10 Lite
  Vm
    mRobot
    UbuntuServer
    Dns
    Metasploit62-Linux
    Black
    EVE-NG
    Kali
    CyberOps_old
    192.168.0.241

Terminal - sudo ~/lab.support.files/scripts/cyberops_topo.py
File Edit View Terminal Tabs Help
sudo ~/lab.support.files/scripts/cyberops_topo.py

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

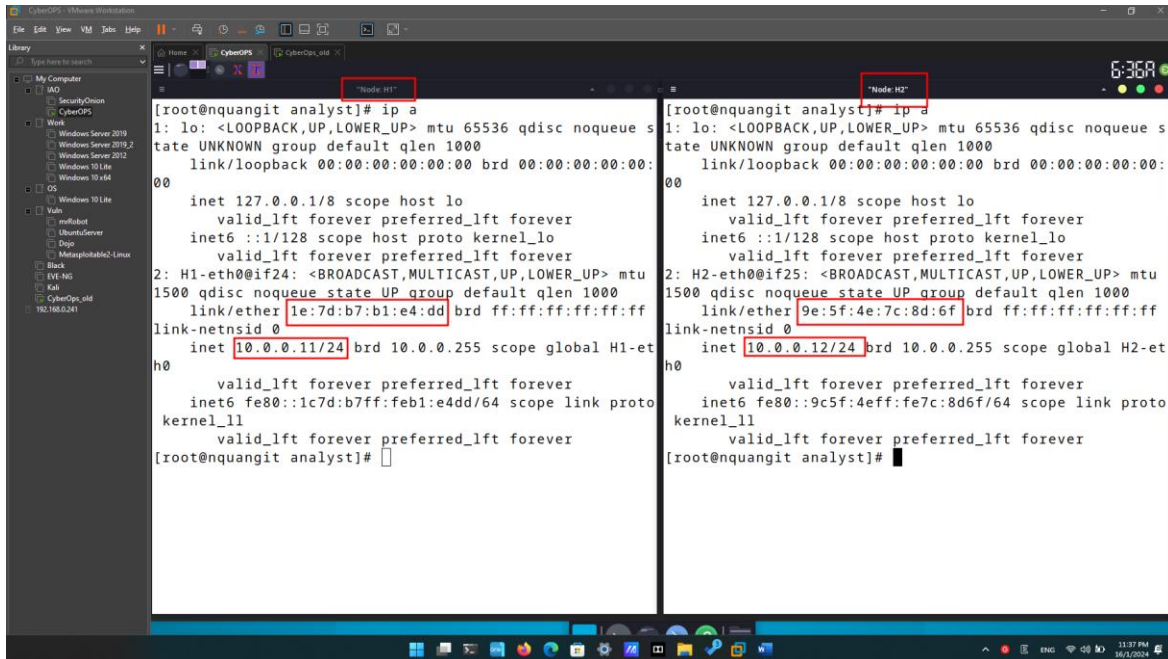
*** Starting 1 switches
s1 ...
*** Routing Table on Router:
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth1
172.16.0.0 0.0.0.0 255.240.0.0 U 0 0 0 R1-eth2

*** Starting CLI:
mininet> xterm H1
mininet> xterm H2
mininet>

[Node H1]
[root@quangit analyst]#

[Node H2]
[root@quangit analyst]#
  
```

- b. At the prompt on **Node: H1**, enter **ip address** to verify the IPv4 address and record the MAC address. Do the same for **Node: H2**. The IPv4 address and MAC address are highlighted below for reference.



```
[root@nquangit analyst]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue s
state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host proto kernel_lo
        valid_lft forever preferred_lft forever
2: H1-eth0@if24: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc noqueue state UP group default qlen 1000
    link/ether 1e:7d:b7:b1:e4:dd brd ff:ff:ff:ff:ff:ff
    link-netnsid 0
    inet 10.0.0.11/24 brd 10.0.0.255 scope global H1-et
h0
        valid_lft forever preferred_lft forever
    inet6 fe80::1c7d:b7ff:feb1:e4dd/64 scope link proto
kernel_ll
        valid_lft forever preferred_lft forever
[root@nquangit analyst]#
```

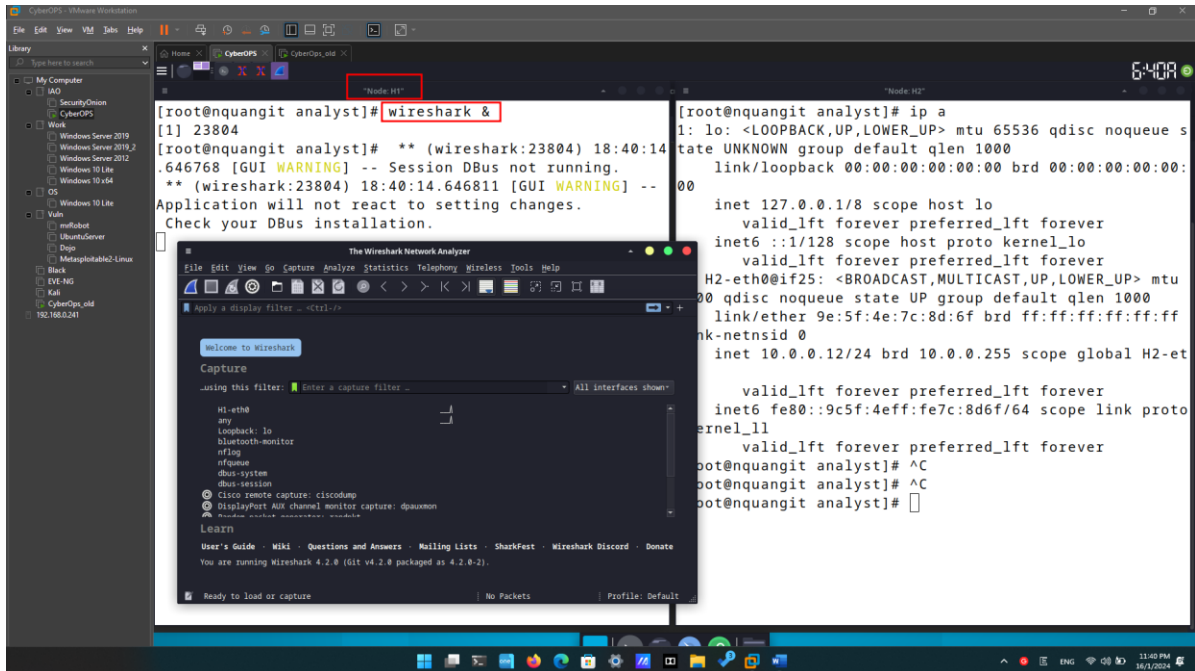
```
[root@nquangit analyst]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue s
state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host proto kernel_lo
        valid_lft forever preferred_lft forever
2: H2-eth0@if25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc noqueue state UP group default qlen 1000
    link/ether 9e:5f:4e:7c:8d:6f brd ff:ff:ff:ff:ff:ff
    link-netnsid 0
    inet 10.0.0.12/24 brd 10.0.0.255 scope global H2-et
h0
        valid_lft forever preferred_lft forever
    inet6 fe80::9c5f:4eff:fe7c:8d6f/64 scope link proto
kernel_ll
        valid_lft forever preferred_lft forever
[root@nquangit analyst]#
```

Host-interface	IP Address	MAC Address
H1-eth0	10.0.0.11/24	1e:7d:b7:b1:e4:dd
H2-eth0	10.0.0.12/24	9e:5f:4e:7c:8d:6f

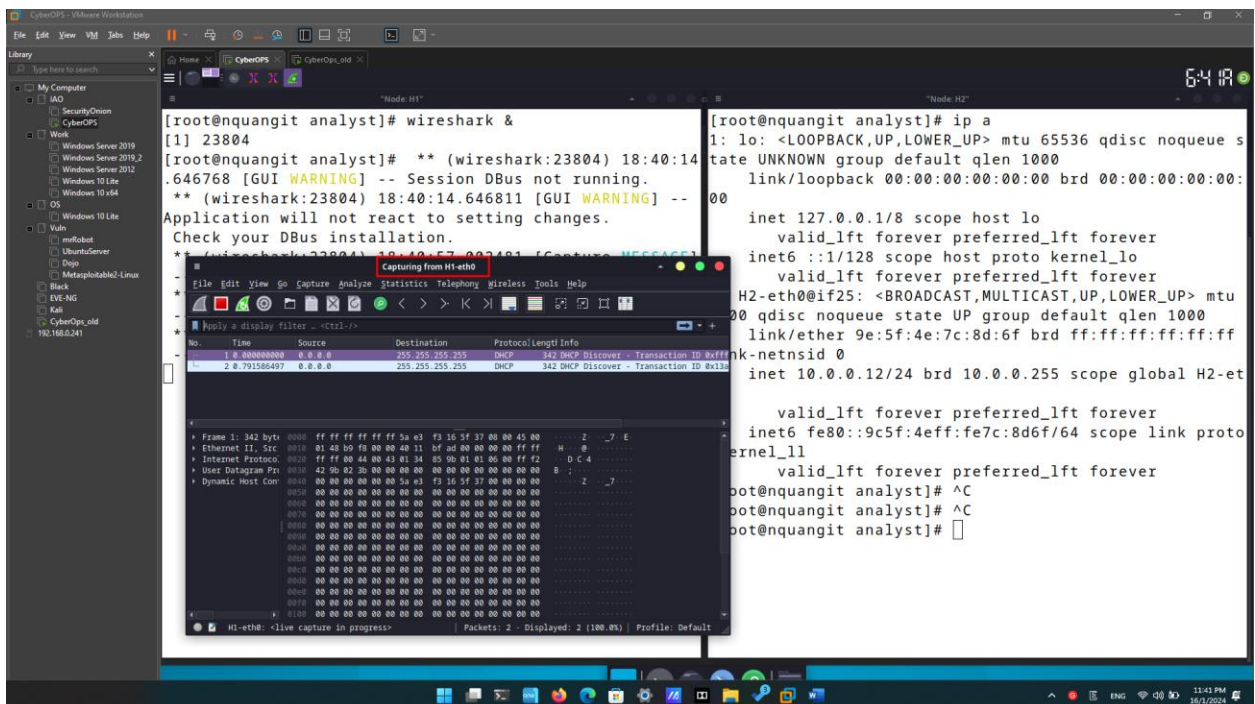
Capture and Analyze ICMP Data in Wireshark

Examine the captured data on the same LAN.

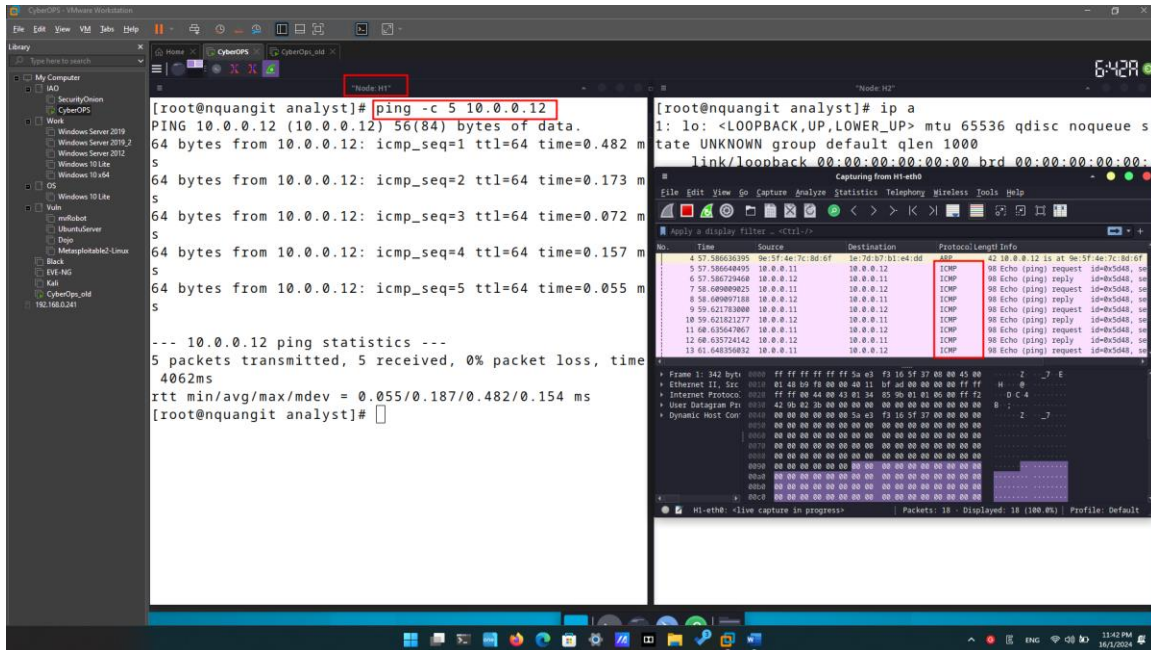
- a. On **Node: H1**, enter **wireshark &** to start Wireshark (The pop-up warning is not important for this lab.). Click **OK** to continue.



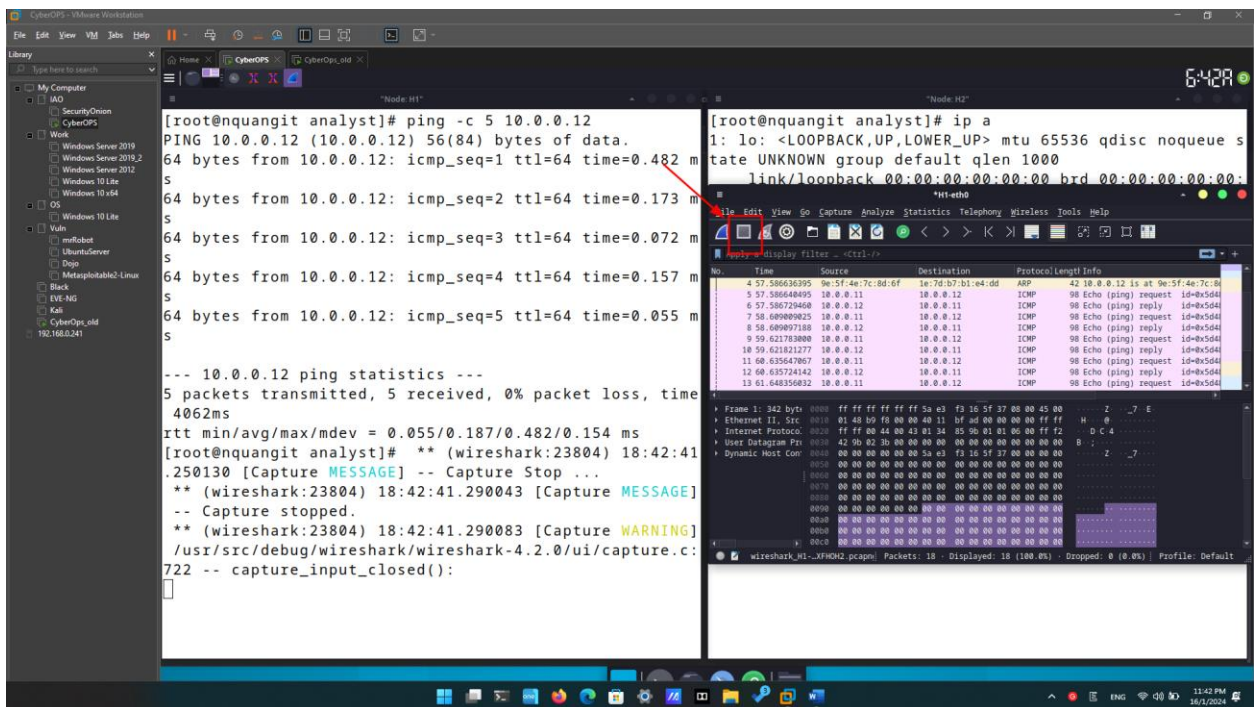
- b. In the Wireshark window, under the **Capture** heading, select the **H1-eth0** interface. Click **Start** to capture the data traffic.



- c. On **Node: H1**, press the Enter key, if necessary, to get a prompt. Then type **ping -c 5 10.0.0.12** to ping H2 five times. The command option **-c** specifies the count or number of pings. The **5** specifies that five pings should be sent. The pings will all be successful.

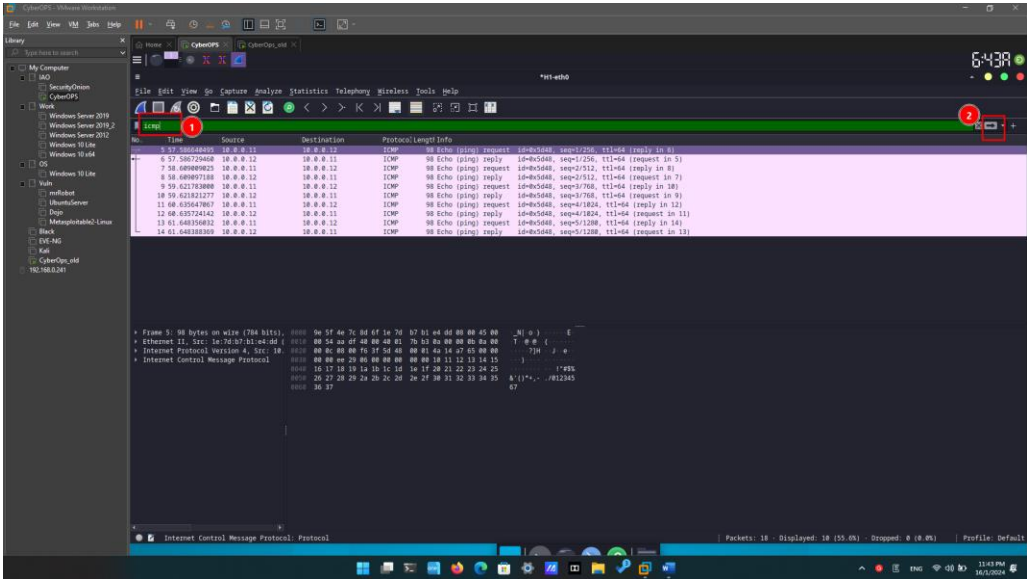
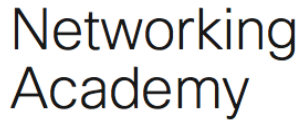


d. Navigate to the Wireshark window, click **Stop** to stop the packet capture.

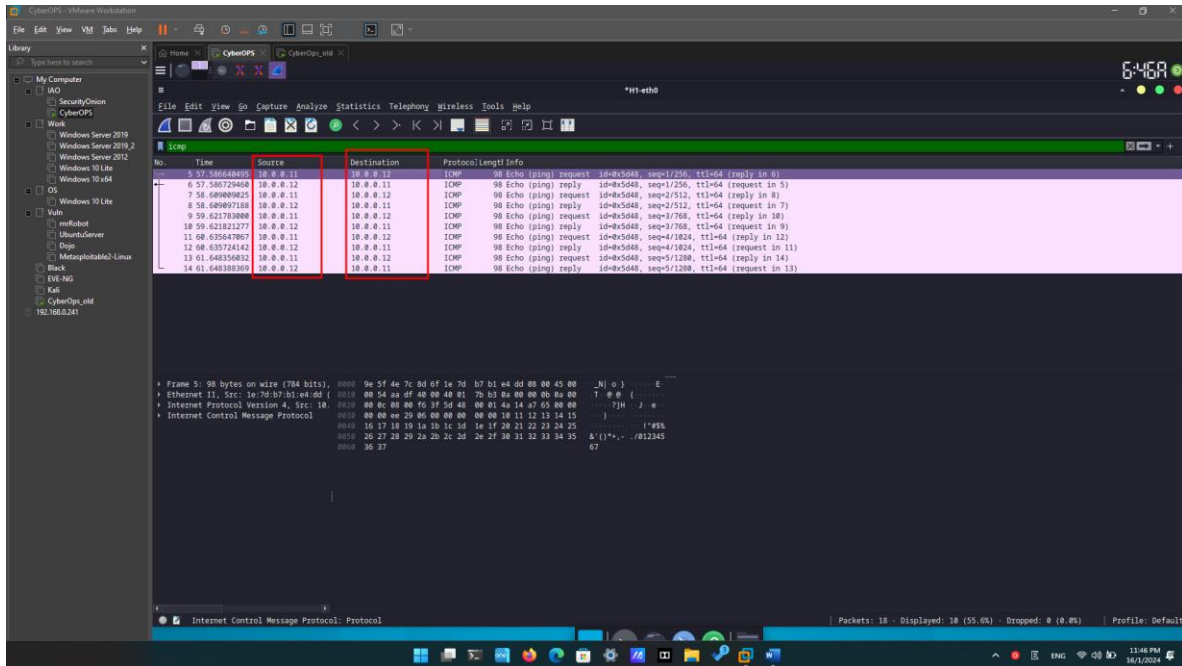


e. A filter can be applied to display only the interested traffic.

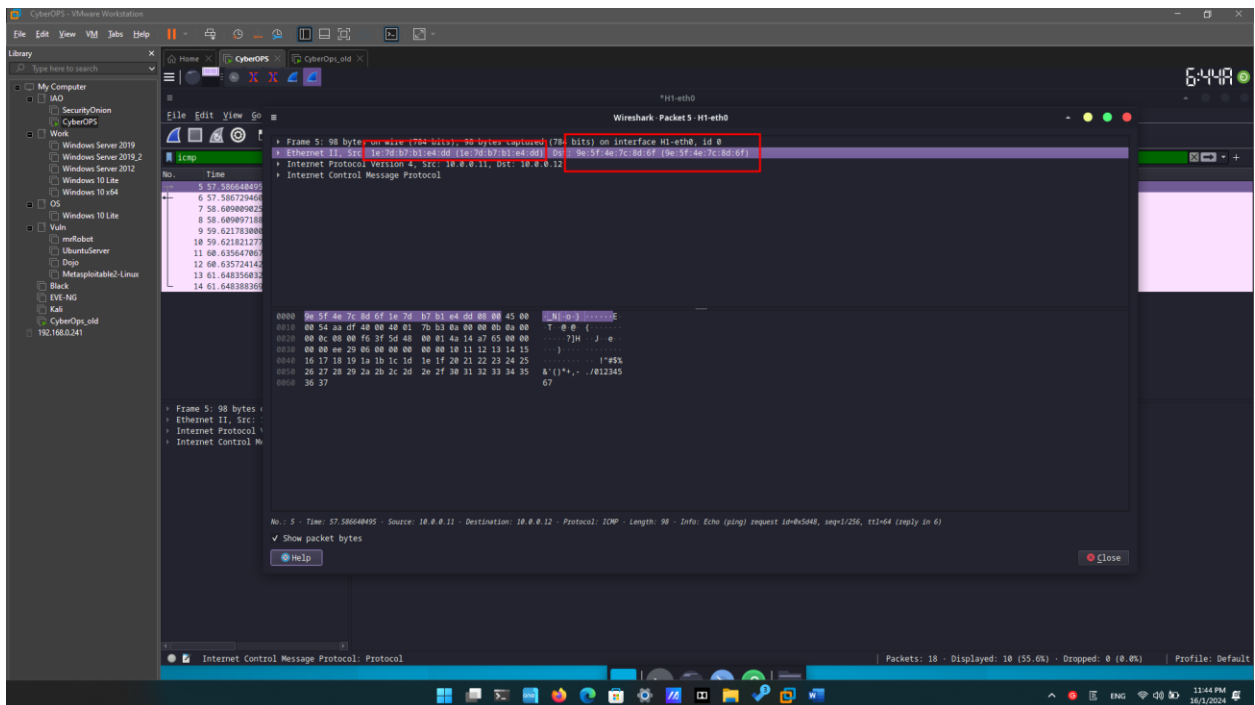
Type **icmp** in the **Filter** field and click **Apply**.



- f. If necessary, click the first ICMP request PDU frames in the top section of Wireshark. Notice that the Source column has H1's IP address, and the Destination column has H2's IP address.



- g. With this PDU frame still selected in the top section, navigate to the middle section. Click the arrow to the left of the Ethernet II row to view the Destination and Source MAC addresses.



Does the Source MAC address match H1's interface?

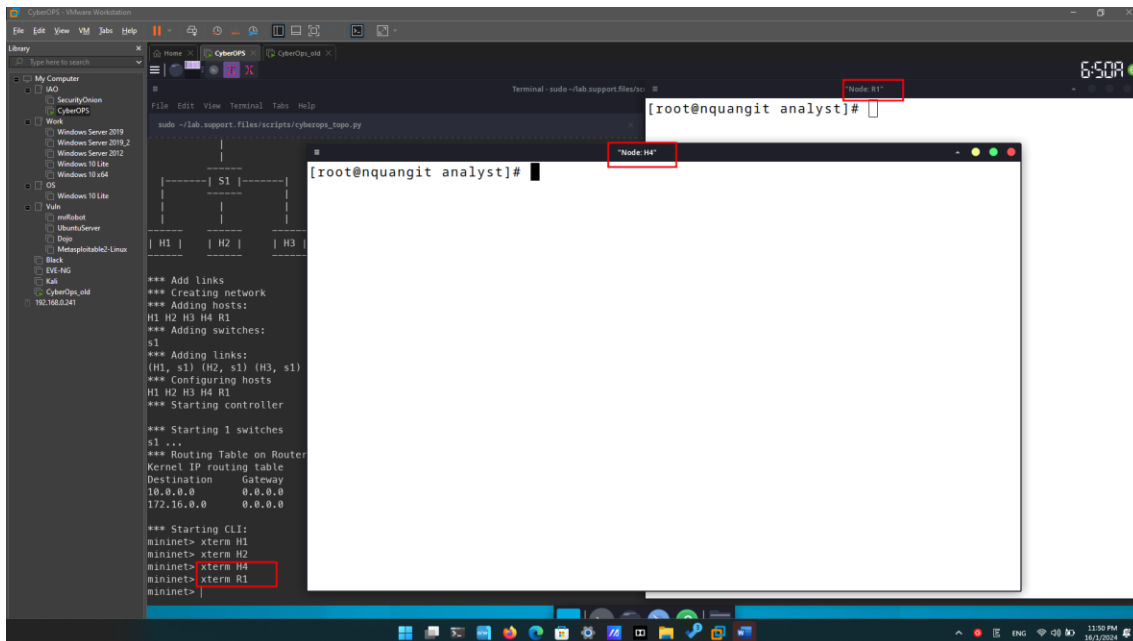
Yes

Does the Destination MAC address in Wireshark match H2's MAC address?

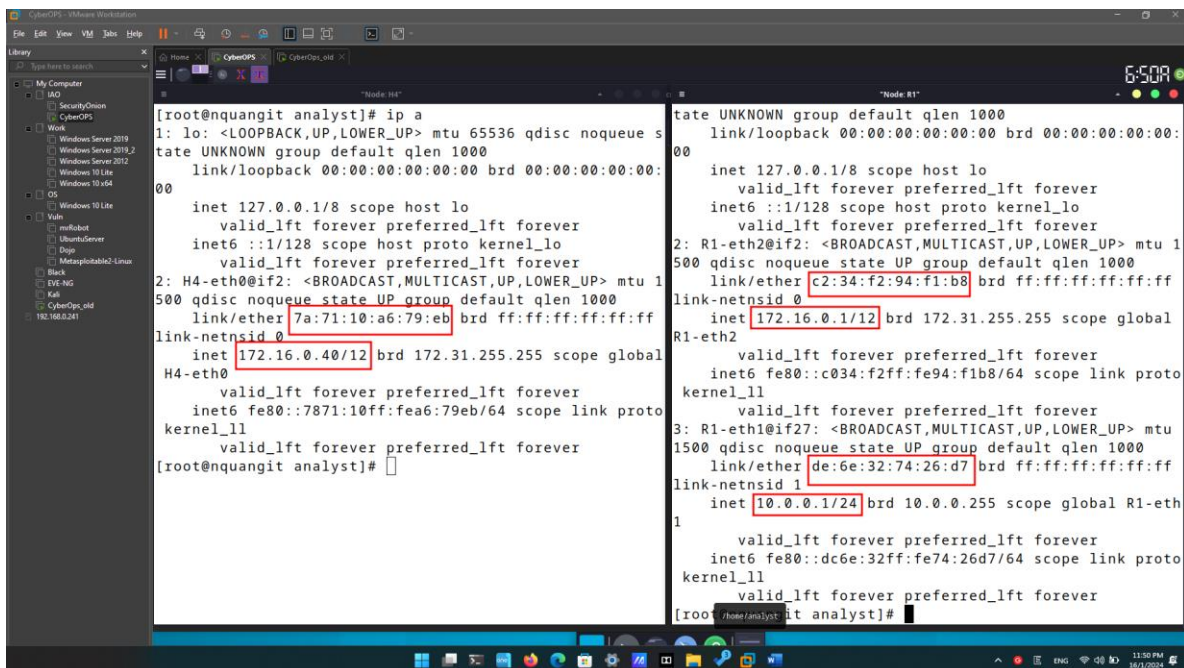
Yes

Examine the captured data on the remote LAN.

- a. At the mininet prompt, start terminal windows on hosts H4 and R1.

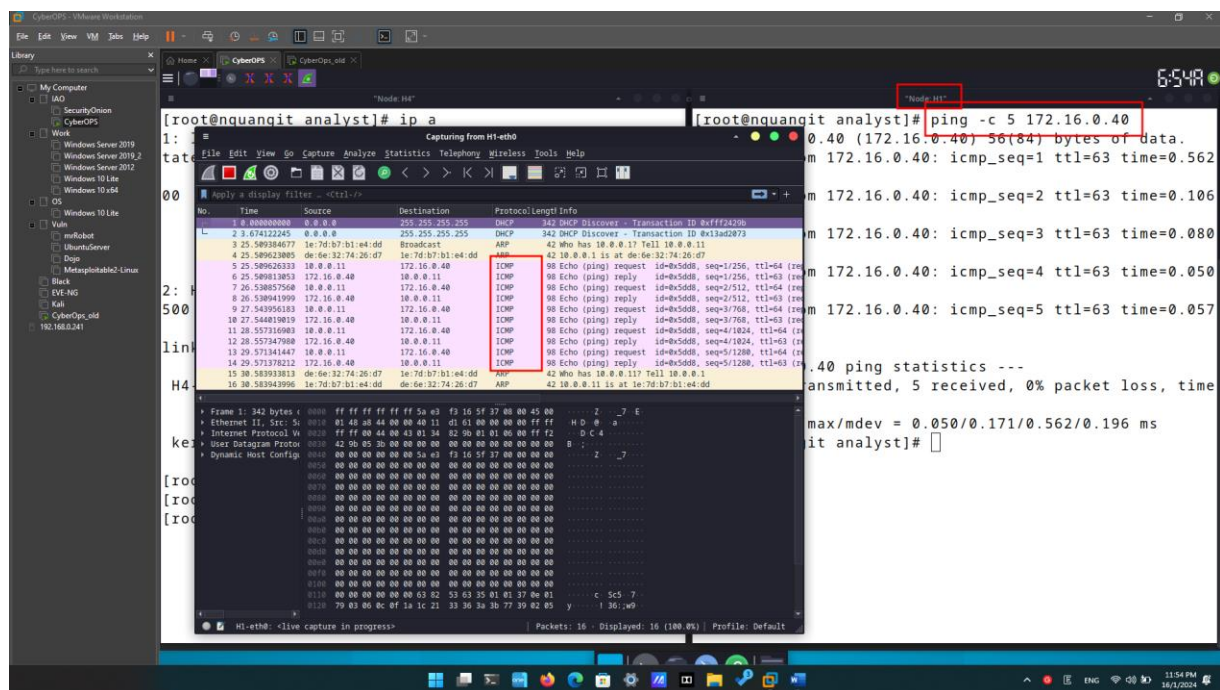


- b. At the prompt on **Node: H4**, enter **ip a** to verify the IPv4 address and record the MAC address. Do the same for the **Node: R1**.

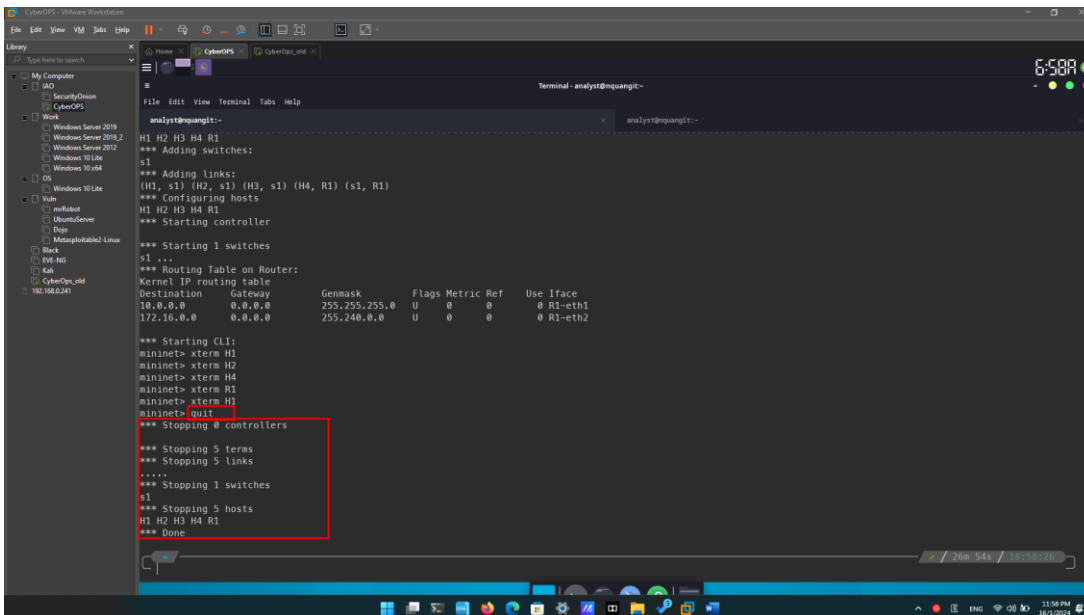


Host-interface	IP Address	MAC Address
H4-eth0	172.16.0.40/12	7a:71:10:a6:79:eb
R1-eth1	10.0.0.1/24	de:6e:32:74:26:d7
R1-eth2	172.16.0.1/12	c2:34:f2:94:f1:b8

- Start a new Wireshark capture on H1 by selecting **Capture > Start**. You can also click the **Start** button or type **Ctrl-E** Click **Continue without Saving** to start a new capture.
- H4 is a simulated remote server. Ping H4 from H1. The ping should be successful.



- Review the captured data in Wireshark. Examine the IP and MAC addresses that you pinged. Notice that the MAC address is for the R1-eth1 interface. List the destination IP and MAC addresses.
 IP address: 172.16.0.40, 10.0.0.11
 MAC address: de:6e:32:74:26:d7, 1e:7d:b7:b1:e4:dd
- In the main CyberOps VM window, enter **quit** to stop Mininet.

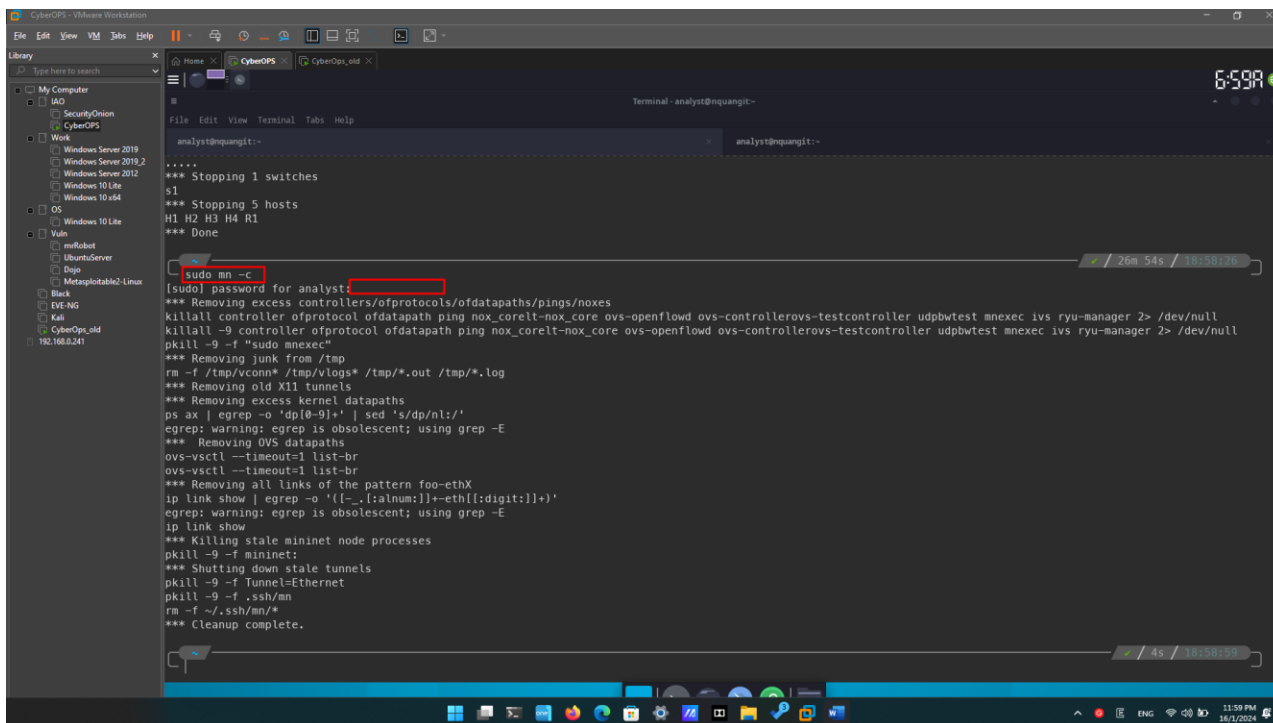


```

analyst@quagga:~$ mininet
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 H2
mininet> xterm H4
mininet> xterm R1
mininet> xterm H1
mininet> quit
*** Stopping 0 controllers
*** Stopping 5 terms
*** Stopping 5 links
....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
  
```

- g. To clean up all the processes that were used by Mininet, enter the **sudo mn -c** command at the prompt.



```

analyst@quagga:~$ sudo mn -c
[sudo] password for analyst:
*** Removing excess controllers/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_corell-nox_core ovs-openflowd ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_corell-nox_core ovs-openflowd ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
kill -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:/'
egrep: warning: egrep is obsolescent; using grep -E
*** 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:]]+'
egrep: warning: egrep is obsolescent; using grep -E
ip link show
*** Killing stale mininet node processes
kill -9 -f mininet:
*** Shutting down stale tunnels
kill -9 -f Tunnel=Ethernet
kill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
  
```