



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE

Applied Computing

Wales Institute of Science and Art (WISA)

Advanced Cybersecurity (ACCB6000)

Lab 2: Security Lab Setup – Feel the Analyst in you

Cisco | **Networking Academy**[®]
Mind Wide Open[™]

What we do?

1. Set up a Lab Environment
2. Feel the Analyst in you
 - a. Browse the options
 - b. Use Mininet
 - c. Trace the route
 - d. Examine the Packets

Complete the lab and submit this document **(save it with your Student_id & Name)** via Moodle before the Next Tutorial Session

Description ¹

Computing power and resources have increased tremendously over the last 10 years. A benefit of having multicore processors and large amounts of RAM is the ability to use virtualization. With virtualization, one or more virtual computers operate inside one physical computer. Virtual computers that run within physical computers are called virtual machines. Virtual machines are often called guests, and physical computers are often called hosts. Anyone with a modern computer and operating system can run virtual machines.

A virtual machine image file has been created for you to install on your computer. In this lab, you will download and import this image file using a desktop virtualization application, such as VirtualBox.

What you need?

- Host computer with a minimum of 8 GB of RAM and 40GB of free disk space
- Internet access

SET UP YOUR OWN LAB**How to do?**

Download and install desktop virtualization software, and also download an image file that can be used to complete labs throughout the course. For this lab, the virtual machine is running Linux.

VMware Player and Oracle VirtualBox are two virtualization programs that you can download and install to support the image file.

In this lab, you will use VirtualBox.

- 1) Navigate to
<http://www.oracle.com/technetwork/serverstorage/virtualbox/downloads/index.html>
Alternatively, download from our **Teams Channel**
- 2) Choose and download the appropriate installation file for your operating system.
- 3) When you have downloaded the VirtualBox installation file, run the installer and accept the default installation settings.
- 4) The image file was created in accordance with the Open Virtualization Format (OVF). OVF is an open standard for packaging and distributing virtual

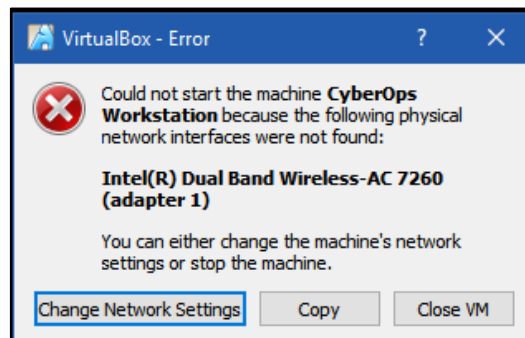
¹ Cyberops Associate – Cisco Networking academy

appliances. An OVF package has several files placed into one directory. This directory is then distributed as an OVA package. This package contains all of the OVF files necessary for the deployment of the virtual machine. The virtual machine used in this lab was exported in accordance with the OVF standard.

- 5) Navigate to the [CyberOps Associates Virtual Machines \(VMs\)](#) page on netacad.com.

Alternatively, download from our **Teams Channel**

- 6) Download the **cyberops_workstation.ova** and **security_onion.ova** image files and note the location of the downloaded VM.
- 7) Open **VirtualBox**. Click **File > Import Appliance...** to import the virtual machine image.
- 8) In the Appliance to import window, specify the location of the .OVA file and click **Next**.
- 9) The Appliance window presents the settings suggested in the OVA archive. Review the default settings and change as necessary. Click **Import** to continue.
- 10) When the import process is complete, you will see the new Virtual Machine added to the VirtualBox inventory in the left panel. The virtual machine is now ready to use.
- 11) Select and start a newly-imported virtual machines. The CyberOps Workstation VM is used as an example in this lab.
- 12) Click the green arrow **Start** button at the top portion of the VirtualBox application window. If you get the following dialog box, click **Change Network Settings** and set your Bridged Adapter. Click the dropdown list next the Name and choose your network adapter (will vary for each computer).



Note: If your network is not configured with DHCP services, click **Change Network Settings** and select **NAT** in the Attached to dropdown box. The network settings can also be access via **Settings** in the Oracle VirtualBox Manager or in the virtual machine menu, select **Devices > Network > Network Settings**. You may need to disable and enable the network adaptor for the change to take effect.

- 13) Click **OK**. A new window will appear, and the virtual machine boot process will start.
- 14) When the boot process is complete, the virtual machine will ask for a username and password. Use the following credentials to log into the virtual machine:

Username: **analyst**

Password: **cyberops**

FEEL THE ANALYST IN YOU

Browse the options

You will be presented with a desktop environment: there is a launcher bar at the bottom, icons on the desktop, and an application menu at the top. Familiarize yourself with the Virtual Machine. The virtual machine you just installed can be used to complete many of the upcoming labs. Familiarize yourself with the icons in the list below

The launcher bar icons are (from left to right):

- Show the desktop
- Terminal application
- File manager application
- Web browser application (Firefox)
- File search tool
- Current user's home directory

All course related applications are located under **Applications Menu > CyberOPs**.

Record your reflection

List the applications in the CyberOps menu.

Open the **Terminal Emulator** application. Type **ip address** at the prompt to determine the IP address of your virtual machine.

What are the IP addresses assigned to your virtual machine?

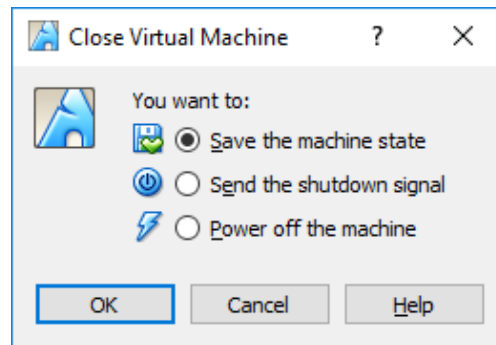
Locate and launch the web browser application.

Can you navigate to your favorite search engine?

When you are done with the VM, you can save the state of VM for future use or shut down the VM.

Closing the VM using GUI:

From the VirtualBox **File** menu, choose **Close**. Click the **Save the machine state** radio button and click **OK**. The next time you start the virtual machine, you will be able to resume working in the operating system in its current state.



The other two options are:

Send the shutdown signal: simulates pressing the power button on a physical computer

Power off the machine: simulates pulling the plug on a physical computer

Closing the VM using CLI:

To shut down the VM using the command line, you can use the menu options inside the VM or enter **sudo shutdown -h now** command in a terminal window and provide the password **cyberops** when prompted.

Rebooting the VM:

If you want to reboot the VM, you can use the menu options inside the VM or enter **sudo reboot** command in a terminal and provide the password **cyberops** when prompted.

Note: You can use the web browser in this virtual machine to research security issues. By using the virtual machine, you may prevent malware from being installed on your computer.

Record your reflection

What are the advantages and disadvantages of using a virtual machine?

Use Mininet

This VM includes a Python script that, when you run it, will set up and configure the devices. You will then have access to four hosts, a switch, and a router inside your one VM. This will allow you to simulate a variety of network protocols and services without having to configure a physical network of devices.

*For example, in this lab you will use the **ping** command between two hosts in the Mininet Topology and capture those pings with Wireshark. You will use Wireshark to capture ICMP data packets.*

Install and Verify the Mininet Topology

In this part, you will use a Python script to set up the Mininet Topology inside the VM. You will then record the IP and MAC addresses for H1 and H2.

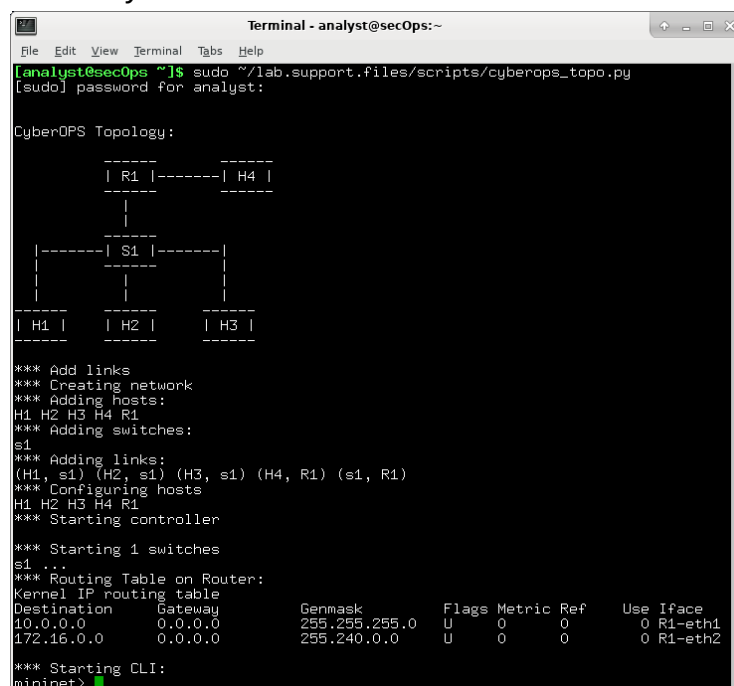
Start and log into your CyberOps Workstation that you have installed.

Username: **analyst** Password: **cyberops**

Run the Python script to install the Mininet Topology.

Open a terminal emulator to start Mininet and enter the following command at the prompt. When prompted, enter **cyberops** as the password.

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



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

[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>
```

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.

*** Starting CLI:

```
mininet> xterm H1
```

```
mininet> xterm H2
```

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@secOps analyst]# ip address
```

<output omitted>

```
2: H1-eth0@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc noqueue state UP group default qlen 1000
```

```
link/ether ba:d4:1d:7b:f3:61 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

```
inet 10.0.0.11/24 brd 10.0.0.255 scope global H1-eth0
```

```
valid_lft forever preferred_lft forever
```

```
inet6 fe80::b8d4:1dff:fe7b:f361/64 scope link
```

```
valid_lft forever preferred_lft forever
```

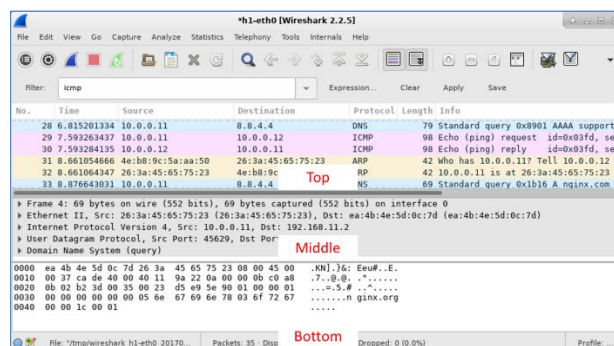
Host-interface	IP Address	MAC Address
H1-eth0		
H2-eth0		

Capture and Analyze ICMP Data in Wireshark

Ping between two hosts in the Mininet and capture ICMP requests and replies in Wireshark. You will also look inside the captured PDUs for specific information. This analysis should help to clarify how packet headers are used to transport data to the destination.

You will examine the data that was generated by the ping requests of your team member's PC. Wireshark data is displayed in three sections:

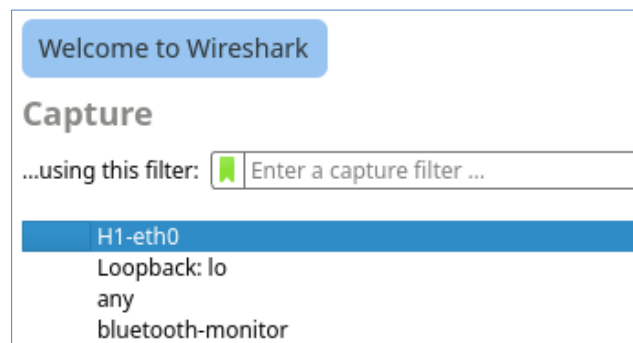
- o The top section displays the list of PDU frames captured with a summary of the IP packet information listed.
- o The middle section lists PDU information for the frame selected in the top part of the screen and separates a captured PDU frame by its protocol layers.
- o The bottom section displays the raw data of each layer. The raw data is displayed in both hexadecimal and decimal form.



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

```
[root@secOps]# wireshark &
[1] 1552
[root@secOps ~]#
** (wireshark:1552): WARNING **: Couldn't connect to accessibility bus:
Failed to connect to socket /tmp/dbus-f0dFz9baYA: Connection refused
Gtk-Message: GtkDialog mapped without a transient parent. This is
discouraged.
```

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



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.

```
[root@secOps analyst]# ping -c 5 10.0.0.12
```

Navigate to the Wireshark window, click **Stop** to stop the packet capture. A filter can be applied to display only the interested traffic. Type **icmp** in the **Filter** field and click **Apply**.

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.

Filter:		icmp				Expression...		Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info				
19	6.791692257	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=1/256, ttl=64 (reply				
20	6.791712977	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=1/256, ttl=64 (reque				
21	7.813333879	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=2/512, ttl=64 (reply				
22	7.813352185	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=2/512, ttl=64 (reque				
23	8.826749959	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=3/768, ttl=64 (reply				
24	8.826773579	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=3/768, ttl=64 (reque				
25	9.839970864	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=4/1024, ttl=64 (repl				
26	9.839991646	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=4/1024, ttl=64 (requ				

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.

```

▶ Frame 19: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▼ Ethernet II, Src: 26:3a:45:65:75:23 (26:3a:45:65:75:23), Dst: 4e:b8:9c:5a:aa:50 (4e:b8:9c:5a:aa:50)
  ▼ Destination: 4e:b8:9c:5a:aa:50 (4e:b8:9c:5a:aa:50)
    Address: 4e:b8:9c:5a:aa:50 (4e:b8:9c:5a:aa:50)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: 26:3a:45:65:75:23 (26:3a:45:65:75:23)
    Address: 26:3a:45:65:75:23 (26:3a:45:65:75:23)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 10.0.0.12
▶ Internet Control Message Protocol

```

Record your reflection

Does the Source MAC address match H1's interface?

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

Note: In the preceding example of a captured ICMP request, ICMP data is encapsulated inside an IPv4 packet PDU (IPv4 header) which is then encapsulated in an Ethernet II frame PDU (Ethernet II header) for transmission on the LAN.

You will ping remote hosts (hosts not on the LAN) and examine the generated data from those pings. You will then determine what is different about this data from the data examined in Part 1.

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

```
mininet> xterm H4
mininet> xterm R1
```

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

```
[root@secOps analyst]# ip address
```

Host-interface	IP Address	MAC Address
H4-eth0		
R1-eth1		
R1-eth2		

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.

```
[root@secOps analyst]# ping -c 5 172.16.0.40
```

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.

Record your reflection

IP address:

MAC address:

In the main CyberOps VM window, enter **quit** to stop Mininet.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 4 terms
*** Stopping 5 links

.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
```

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

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

```
[sudo] password for analyst:
```

```
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
*** Cleanup complete.
```

Traceroute

Open a terminal window in the VM to ping a remote server, such as www.cisco.com.

```
[analyst@secOps ~]$ ping -c 4 www.cisco.com
```

```
PING e2867.dsca.akamaiedge.net (184.24.123.103) 56(84) bytes of data.  
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com  
(184.24.123.103): icmp_seq=1 ttl=59 time=13.0 ms  
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com  
(184.24.123.103): icmp_seq=2 ttl=59 time=12.5 ms  
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com  
(184.24.123.103): icmp_seq=3 ttl=59 time=14.9 ms  
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com  
(184.24.123.103): icmp_seq=4 ttl=59 time=11.9 ms
```

```
--- e2867.dsca.akamaiedge.net ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 11.976/13.143/14.967/1.132 ms
```

The first output line displays the Fully Qualified Domain Name (FQDN) e2867.dsca.akamaiedge.net. This is followed by the IP address 184.24.123.103. Cisco hosts the same web content on different servers throughout the world (known as mirrors). Therefore, depending upon where you are geographically, the FQDN and the IP address will be different.

Four pings were sent and a reply was received from each ping. Because each ping received a response, there was 0% packet loss. On average, it took 3005 ms (3005 milliseconds) for the packets to cross the network. A millisecond is 1/1,000th of a second. Your results will likely be different.

Tracing a Route to a Remote Server Using Traceroute

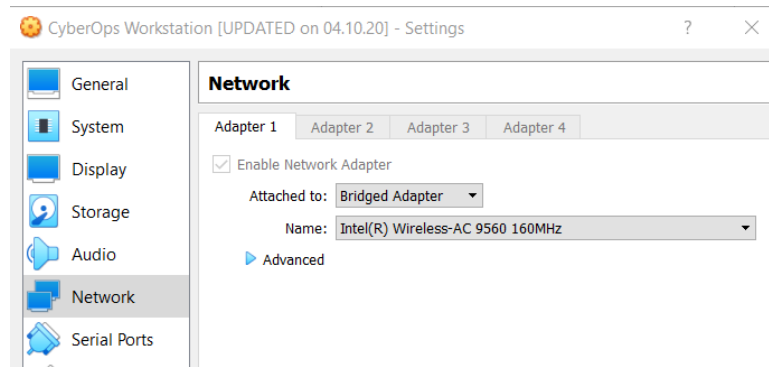
Now that basic reachability has been verified by using the ping tool, it is helpful to look more closely at each network segment that is crossed.

Routes traced can go through many hops and a number of different Internet Service Providers (ISPs), depending on the size of your ISP and the location of the source and destination hosts. Each “hop” represents a router. A router is a specialized type of computer used to direct traffic across the internet. Imagine taking an automobile trip across several countries using many highways. At different points in the trip you come to a fork in the road in which you have the option to select from several different highways. Now further imagine that there is a device at each fork in the road that directs you to take the correct highway to your final destination. That is what a router does for packets on a network.

Because computers talk in binary or hexadecimal numbers, rather than words, routers are uniquely identified using IP addresses. The **traceroute** tool shows you what path through the network a packet of information takes to reach its final

destination. The **traceroute** tool also gives you an idea of how fast traffic is going on each segment of the network. Packets are sent to each router in the path, and the return time is measured in milliseconds.

Note: CyberOps Workstation VM network settings may need to be set to bridged adapter if you are not getting any traceroute results. To check your network settings, go to: **Machine > Settings**, select **Network**, the tab Adapter 1, Attached to: **Bridged Adapter**.



To do this, the **traceroute** tool is used.

At the terminal prompt, type **traceroute www.cisco.com**.

```
[analyst@secOps ~]$ traceroute www.cisco.com
traceroute to www.cisco.com (184.24.123.103), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1) 6.527 ms 6.783 ms 6.826 ms
 2 10.39.176.1 (10.39.176.1) 27.748 ms 27.533 ms 27.480 ms
 3 100.127.65.250 (100.127.65.250) 27.864 ms 28.570 ms 28.566 ms
 4 70.169.73.196 (70.169.73.196) 29.063 ms 35.025 ms 33.976 ms
 5 fed1bbrj01.xe110.0.rd.sd.cox.net (68.1.0.155) 39.101 ms 39.120 ms
   39.108 ms
 6 a184-24-123-103.deploy.static.akamaitechnologies.com (184.24.123.103)
   38.004 ms 13.583 ms 13.612 ms
```

If you would like to save the traceroute output to a text file for later review, use the right carat (>) and the desired filename to save the output in the present directory. In this example, the traceroute output is saved in the /home/analyst/cisco-traceroute.txt file.

```
[analyst@secOps ~]$ traceroute www.cisco.com > cisco-traceroute.txt
```

You can now enter the **cat cisco-traceroute.txt** command to view the output of the trace stored in the text file.

Perform and save the traceroute results for one of the following websites. These are the Regional Internet Registry (RIR) websites located in different parts of the world:

Africa: **www.afrinic.net**

Australia: **www.apnic.net**

Europe: **www.ripe.net**

South America: **www.lacnic.net**

Note: Some of these routers along the route may not respond to traceroute.

Trace a Route to a Remote Server Using Web-Based Traceroute Tool

Open a web browser in the VM and search for a visual traceroute tool that you can use in the web browser. Try going to the following website:
<https://gsuite.tools/traceroute>

Enter any website you wish. **Example: www.cisco.com** and press **Trace**.

Note: If you get the error “SEC_ERROR_OCSP_FUTURE_RESPONSE” in Firefox then the CyberOps Workstation clock/time is incorrect. To fix the time enter the following command to update the clock/time then refresh the web browser and enter the visual trace:

```
[analyst@secOps ~]$ sudo ntpd -qq
```

Record your reflection

Review the geographical locations of the responding hops. What did you observe regarding the path?

How is the traceroute different when going to www.cisco.com or other websites from the terminal (see Part 2) rather than from the online website? (Your results may vary depending upon where you are located geographically, and which ISP is providing connectivity to your school.)

Examine the Packets

The Wireshark capture below shows the packets generated by a ping being issued from a PC host to its default gateway. A filter has been applied to Wireshark to view the ARP and ICMP protocols only. The session begins with an ARP query for the MAC address of the gateway router, followed by four ping requests and replies.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.518367	IntelCor_62:62:6d	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.6
11	0.523508	Netgear_ea:b1:7a	IntelCor_62:62:6d	ARP	42	192.168.1.1 is at 80:37:73:ea:b1:7a
12	0.523633	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=85...
14	0.530258	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=85...
17	0.857735	Microchi_32:b7:3d	Broadcast	ARP	60	Who has 192.168.1.1? Tell 192.168.1.3

Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0		
Ethernet II, Src: IntelCor_62:62:6d (f4:8c:50:62:62:6d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)		
Destination: Broadcast (ff:ff:ff:ff:ff:ff)		
Source: IntelCor_62:62:6d (f4:8c:50:62:62:6d)		
Type: ARP (0x0806)		
Address Resolution Protocol (request)		

Offset	Hex	ASCII
0000	ff ff ff ff ff ff f4 8c 50 62 62 6d 08 06 00 01	 Pbbm....
0010	08 00 06 04 00 01 f4 8c 50 62 62 6d c0 a8 01 06	 Pbbm....
0020	00 00 00 00 00 00 c0 a8 01 01	

The following table takes the first frame in the Wireshark capture and displays the data in the Ethernet II header fields.

Field	Value	Description						
Preamble	Not shown in capture	This field contains synchronizing bits, processed by the NIC hardware.						
Destination Address	Broadcast (ff:ff:ff:ff:ff:ff)	Layer 2 addresses for the frame. Each address is 48 bits long, or 6 octets, expressed as 12 hexadecimal digits, 0-9,A-F. A common format is 12:34:56:78:9A:BC. The first six hex numbers indicate the manufacturer of the network interface card (NIC), the last six hex numbers are the serial number of the NIC. The destination address may be a broadcast, which contains all ones, or a unicast. The source address is always unicast.						
Source Address	IntelCor_62:62:6d (f4:8c:50:62:62:6d)							
Frame Type	0x0806	For Ethernet II frames, this field contains a hexadecimal value that is used to indicate the type of upper-layer protocol in the data field. There are numerous upper-layer protocols supported by Ethernet II. Two common frame types are: <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0800</td><td>IPv4 Protocol</td></tr><tr><td>0x0806</td><td>Address resolution protocol (ARP)</td></tr></table>	Value	Description	0x0800	IPv4 Protocol	0x0806	Address resolution protocol (ARP)
Value	Description							
0x0800	IPv4 Protocol							
0x0806	Address resolution protocol (ARP)							
Data	ARP	Contains the encapsulated upper-level protocol. The data field is between 46 – 1,500 bytes.						
FCS	Not shown in capture	Frame Check Sequence, used by the NIC to identify errors during transmission. The value is computed by the sending machine, encompassing frame addresses, type, and data field. It is verified by the receiver.						

Record your reflection

What is significant about the contents of the destination address field?

Why does the PC send out a broadcast ARP prior to sending the first ping request?

What is the MAC address of the source in the first frame?

What is the Vendor ID (OUI) of the Source's NIC?

What portion of the MAC address is the OUI?

What is the Source's NIC serial number?

Use Wireshark to Capture and Analyze Ethernet Frames

Start and log into your CyberOps Workstation VM using the following credentials:

Username: **analyst** Password: **cyberops**

Open a terminal emulator to start mininet and enter the following command at the prompt. When prompted, enter **cyberops** as the password.

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

At the mininet prompt, start terminal windows on host H3.

```
*** Starting CLI:  
mininet> xterm H3
```

At the prompt on Node: h3, enter **ip address** to verify the IPv4 address and record the MAC address.

Host-interface	IP Address	MAC Address
H3-eth0		

At the prompt on Node: H3, enter **netstat -r** to display the default gateway information.

```
[root@secOps ~]# netstat -r  
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	10.0.0.1	0.0.0.0	UG	0	0		H3-eth0
10.0.0.0	0.0.0.0	255.255.255.0	U	0	0		H3-eth0

Record your reflection

What is the IP address of the default gateway for the host H3?

In the terminal window for Node: H3, enter **arp -n** to display the content of the ARP cache.

```
[root@secOps analyst]# arp -n
```

If there is any existing ARP information in the cache, clear it by enter the following command: **arp -d IP-address**. Repeat until all the cached information has been cleared.

```
[root@secOps analyst]# arp -n
```

Address	HWtype	HWaddress	Flags	Mask	Iface
10.0.0.11	ether	5a:d0:1d:01:9f:be	C		H3-eth0

```
[root@secOps analyst]# arp -d 10.0.0.11
```

Address	HWtype	HWaddress	Flags	Mask	Iface
10.0.0.11	(incomplete)		C		H3-eth0

In the terminal window for Node: H3, open Wireshark and start a packet capture for H3-eth0 interface.

```
[root@secOps analyst]# wireshark-gtk &
```

Ping H1 from H3. From the terminal on H3, ping the default gateway and stop after send 5 echo request packets.

```
[root@secOps analyst]# ping -c 5 10.0.0.1
```

After the ping is completed, stop the Wireshark capture. Apply the **icmp** filter to the captured traffic so only ICMP traffic is shown in the results.

In the Packet List pane (top section), click the first frame listed. You should see **Echo (ping) request** under the **Info** heading. This should highlight the line blue.

Examine the first line in the Packet Details pane (middle section). This line displays the length of the frame;

The second line in the Packet Details pane shows that it is an Ethernet II frame. The source and destination MAC addresses are also displayed.

Record your reflection

What is the MAC address of the PC's NIC?

What is the default gateway's MAC address?

You can click the arrow at the beginning of the second line to obtain more information about the Ethernet II frame.

What type of frame is displayed?

The last two lines displayed in the middle section provide information about the data field of the frame. Notice that the data contains the source and destination IPv4 address information.

What is the source IP address?

What is the destination IP address?

*You can click any line in the middle section to highlight that part of the frame (hex and ASCII) in the Packet Bytes pane (bottom section). Click the **Internet Control Message Protocol** line in the middle section and examine what is highlighted in the Packet Bytes pane.*

```

▶ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▼ Ethernet II, Src: 42:28:b2:24:e0:cb (42:28:b2:24:e0:cb), Dst: 92:66:62:f0:14:21 (92:66:62:f0:14:21)
  ▶ Destination: 92:66:62:f0:14:21 (92:66:62:f0:14:21)
  ▶ Source: 42:28:b2:24:e0:cb (42:28:b2:24:e0:cb)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.0.0.13, Dst: 10.0.0.1
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xa77b [correct]
  [Checksum Status: Good]
  Identifier (BE): 3711 (0x0e7f)
  Identifier (LE): 32526 (0x7f0e)
  Sequence number (BE): 1 (0x0001)
  Sequence number (LE): 256 (0x0100)
  [Response frame: 4]
  Timestamp from icmp data: Apr 26, 2017 20:45:51.878857000 EDT
  [Timestamp from icmp data (relative): 0.000210324 seconds]
▼ Data (48 bytes)
  Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f...
  [Length: 48]
0000  92 66 62 f0 14 21 42 28 b2 24 e0 cb 08 00 45 00 .fb..!B( .$....E.
0010  00 54 58 66 40 00 40 01 ce 35 0a 00 00 0d 0a 00 .TXf@.@. .5.....
0020  00 01 08 00 a7 7b 0e 7f 00 01 3f 3f 01 59 09 69 ...{...???.Y.1
0030  0d 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060  36 37 67

```

Click the **Start Capture** icon to start a new Wireshark capture. You will receive a popup window asking if you would like to save the previous captured packets to a file before starting a new capture. Click **Continue without Saving**.

In the terminal window of Node: H3, send 5 echo request packets to 172.16.0.40.

Stop capturing packets when the pings are completed.

Examine the new data in the packet list pane of Wireshark.

Record your reflection

In the first echo (ping) request frame, what are the source and destination MAC addresses?

Source:

Destination:

What are the source and destination IP addresses contained in the data field of the frame?

Source:

Destination:

Wireshark does not display the preamble field of a frame header. What does the preamble contain?