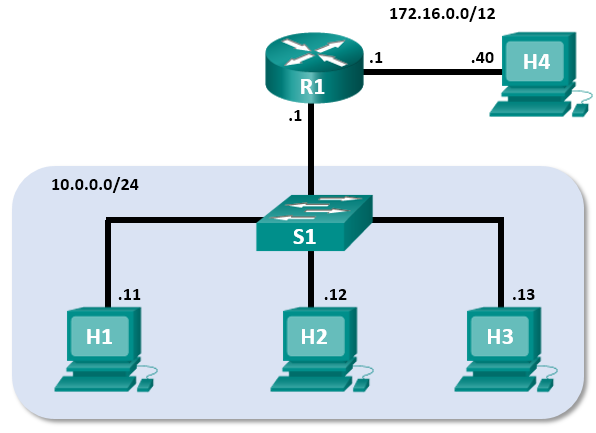
Práctica de laboratorio: Introducción a Wireshark (versión para el instructor)

**Nota para el instructor**: Los elementos con color de fuente rojo o resaltados en gris indican texto que aparece solo en la copia del instructor.

# Topología Mininet



# Objetivos

Parte 1: Instalar y verificar la topología de Mininet

Parte 2: Capturar y analizar datos de ICMP en Wireshark

# Antecedentes / Escenario

La VM CyberOps incluye un script de Python que, cuando se ejecuta, instala y configura los dispositivos que se muestran en la figura de arriba. Entonces podrán acceder a cuatro hosts, a un switch y a un router que se encuentra dentro de sus VM únicas. Esto les permitirá simular diversos protocolos y servicios de red sin tener que configurar una red física de dispositivos. Por ejemplo: en esta práctica de laboratorio utilizarán el comando **ping** entre dos hosts en la topología de Mininet y capturarán esos pings con Wireshark.

Wireshark es un analizador de protocolos de software o una aplicación “husmeadora de paquetes” que se utiliza para solución de problemas de red, análisis, desarrollo de protocolos y software y educación. Mientras los flujos de datos transitan por la red, el husmeador “captura” cada unidad de datos del protocolo (Protocol Data Unit, PDU) y puede decodificar y analizar su contenido de acuerdo a la RFC correcta o a otras especificaciones.

Wireshark es una herramienta útil para cualquier persona que trabaje con redes y necesite capacidades de análisis de datos y solución de problemas. Utilizarán Wireshark para capturar paquetes de datos ICMP.

# Recursos necesarios

* Máquina virtual CyberOps Workstation

**Nota para el instructor**: El uso de un analizador de protocolos de paquetes como Wireshark se puede considerar una infracción de la política de seguridad del lugar de estudios. Se recomienda obtener autorización antes de realizar esta práctica de laboratorio. Si el uso de un programa detector de paquetes como Wireshark resulta ser un problema, el instructor puede asignar la práctica de laboratorio como tarea para el hogar o hacer una demostración.

# Instrucciones

## Instalar y verificar la topología de Mininet

En esta parte utilizarán un script de Python para configurar la topología de Mininet dentro de la VM CyberOps. Luego, registrarán las direcciones IP y MAC correspondientes a H1 y a H2.

### Verificar las direcciones de las interfaces de sus PC

Inicien sus VM CyberOps Workstation que han instalado en una práctica anterior e inicien sesión con las siguientes credenciales:

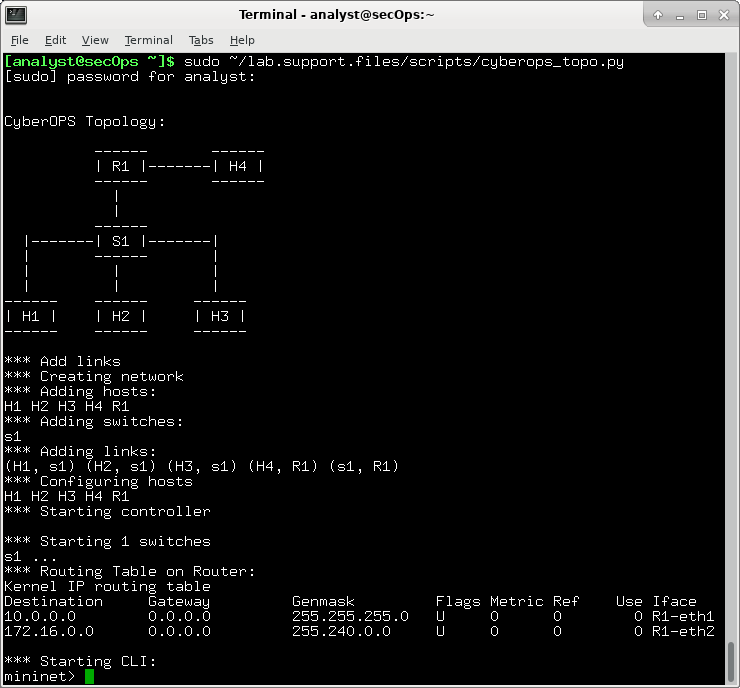
Nombre de usuario: **analyst** Contraseña: **cyberops**

### Ejecutar el script de Python para instalar la topología de Mininet

Abrir un emulador de terminales para iniciar Mininet e introducir el siguiente comando. Cuando el sistema se lo solicite, introduzca **cyberops** como la contraseña.

[analyst@secOps ~]$ **sudo ~/lab.support.files/scripts/cyberops\_topo.py**

[sudo] contraseña para analyst:



### Registrar las direcciones IP y MAC correspondientes a H1 y a H2.

* + 1. En el cursor de mininet, inicien ventanas del terminal en los hosts H1 y H2. Se abrirán ventanas aparte para estos hosts. Cada host tendrá su configuración independiente para la red, incluidas las direcciones IP y MAC únicas.

\*\*\* Starting CLI:

mininet> **xterm H1**

mininet> **xterm H2**

* + 1. En el prompt en **Node:H1**, introducir **ip adress** para verificar la dirección IPv4 y registrar la dirección MAC. Hagan lo mismo en **Node: H2**. Tanto la dirección IPv4 como la MAC están resaltadas a continuación a modo de referencia.

[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-interfaz | Dirección IP | Dirección MAC |
| H1-eth0 | 10.0.0.11 | Las respuestas pueden variar. ba:d 4:1 d:7b:f 3:61 como ejemplo |
| H2-eth0 | 10.0.0.12 | Las respuestas pueden variar. 4e:b 8:9 c:5a:aa:50 como ejemplo. |

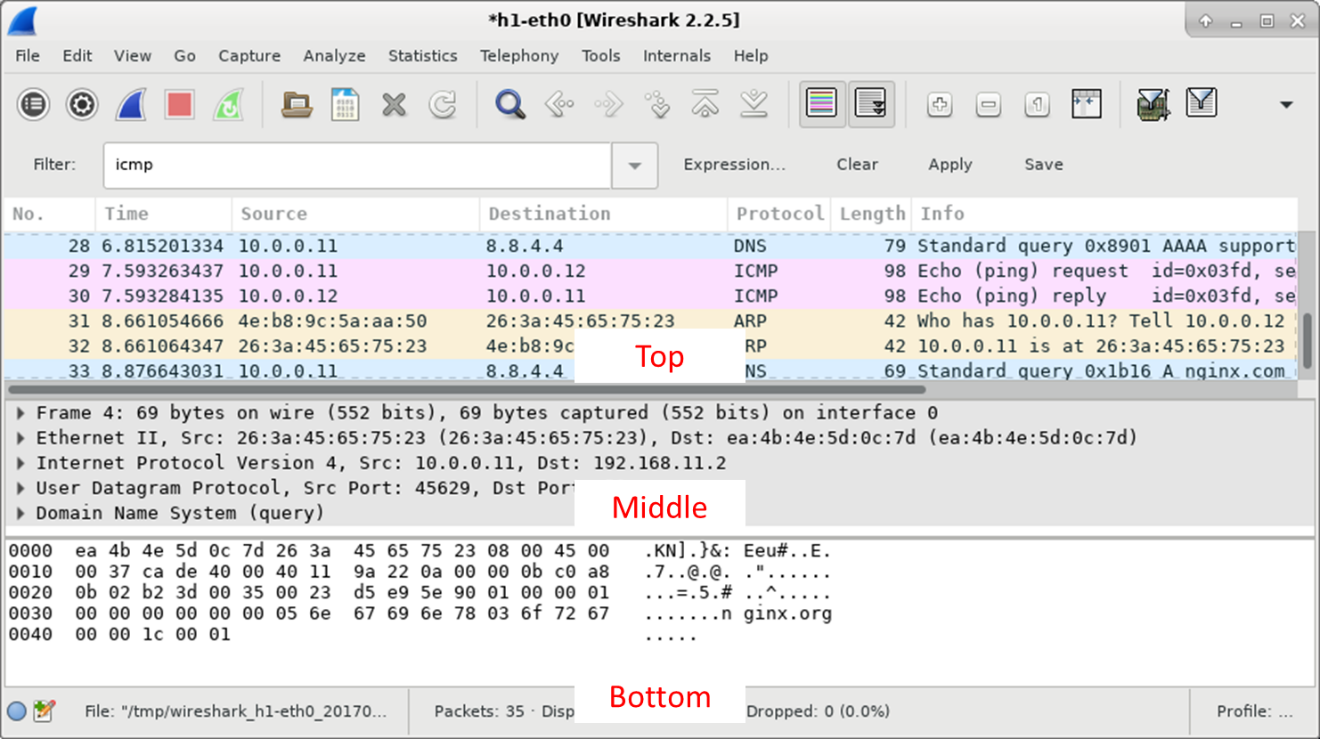
## Captura y análisis de datos de ICMP en Wireshark

En esta parte harán pings entre dos hosts en la Mninet y capturarán solicitudes y respuestas ICMP en Wireshark. También verán el contenido de las PDU capturadas para obtener información específica. Este análisis debería ser útil para aclarar de qué manera se utilizan los encabezados de paquetes para transportar datos al destino.

### Examinar los datos capturados en la misma red LAN

En este paso examinarán los datos que se generaron mediante las solicitudes de ping de la PC del miembro de su equipo. Los datos de Wireshark se muestra en tres secciones:

* La sección superior muestra la lista de tramas de PDU capturadas con un resumen de la información de paquetes IPv4.
* La sección del medio muestra la información de PDU de la trama seleccionada en la parte superior de la pantalla y separa una trama de PDU capturada por las capas del protocolo.
* La sección inferior muestra los datos de cada capa sin formato. Los datos sin procesar se muestran en formatos hexadecimal y decimal.



* + - 1. En **Node:H1**, introducir **wireshark &** para iniciar Wireshark (La advertencia emergente no es importante en esta práctica). Haga clic en **Aceptar** para continuar.

[root@secOps]# **wireshark &**

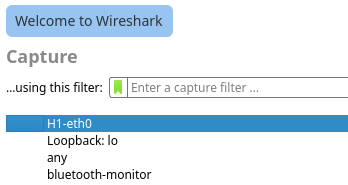
[1] 1552

[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.

* + - 1. En la ventana de Wireshark, en el encabezado **Capture** (Capturar), seleccionen la interfaz **H1-eth0**. Hagan clic en **Start** (Comenzar) para capturar el tráfico de datos.



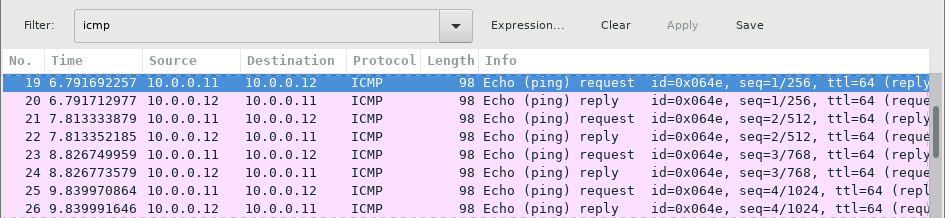
* + - 1. En **Node: H1**, presionen la tecla Intro (si es necesario) para ver un cursor. Luego, escriban **ping -c 5 10.0.0.12** para hacer ping a H2 cinco veces. La opción de comando **-c** especifica la cantidad o el número de pings. El **5** especifica que se deben enviar cinco pings. Todos los pings se realizarán correctamente.

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

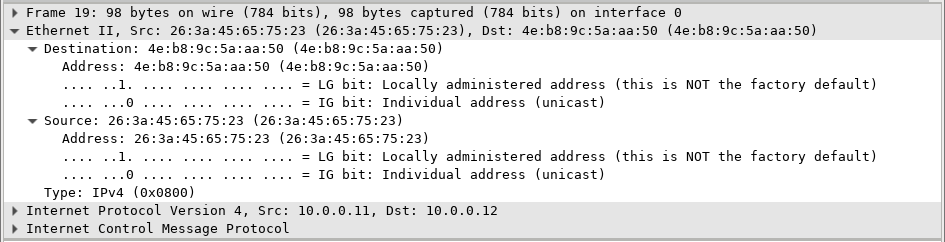
* + - 1. Diríjanse a la ventana de Wireshark y hagan clic en **Stop** (Detener) para detener la captura de paquetes.
      2. Se puede aplicar un filtro para mostrar solamente el tráfico de interés.

Escriban **icmp** en el campo **Filter** (Filtro) y hagan clic en **Apply** (Aplicar).

* + - 1. Si es necesario, hagan clic en las primeras tramas de PDU de la solicitud ICMP en la sección superior de Wireshark. Observen que la columna Source (Origen) tiene la dirección IP de H1, y que la columna Destination (Destino) tiene la dirección IP de H2.



* + - 1. Con esta trama de PDU aún seleccionada en la sección superior, navegue hasta la sección media. Haga clic en la flecha que se encuentra a la izquierda de la fila Ethernet II para ver las direcciones MAC de origen y de destino.



#### Pregunta:

¿La dirección MAC de origen coincide con la interfaz de H1?.

Sí

¿La dirección MAC de destino en Wireshark coincide con la dirección MAC de H2?

Sí

**Nota**: En el ejemplo anterior de una solicitud de ICMP capturada, los datos ICMP se encapsulan dentro de una PDU del paquete IPv4 (encabezado de IPv4), que luego se encapsula en una PDU de trama de Ethernet II (encabezado de Ethernet II) para la transmisión en la LAN.

### Examinar los datos capturados en la red LAN remota

Harán pings a hosts remotos (hosts que no están en la LAN) y examinarán los datos generados a partir de esos pings. Luego, determinarán las diferencias entre estos datos y los datos examinados en la parte 1.

* + - 1. En el cursor de mininet, inicien ventanas del terminal en los hosts H4 y R1.

mininet> **xterm H4**

mininet> **xterm R1**

* + - 1. En el prompt en **Node: H4**, introducir **ip address** para verificar la dirección IPv4 y registrar la dirección MAC. Hagan lo mismo en **Node: R1**.

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

|  |  |  |
| --- | --- | --- |
| Host-interfaz | Dirección IP | Dirección MAC |
| H4-eth0 | 172.16.0.40 | Las respuestas pueden variar. |
| R1-eth1 | 10.0.0.1 | Las respuestas pueden variar. |
| R1-eth2 | 172.16.0.1 | Las respuestas pueden variar. |

* + - 1. Inicien una captura de Wireshark nueva en H1; para ello, seleccionen **Capture > Start** (Captura > Comenzar). También pueden hacer clic en el botón **Start** o presionar **Ctrl-E**. Hagan clic en **Continue without Saving** (Continuar sin guardar) para iniciar una captura nueva.
      2. H4 es un servidor remoto simulado. Hagan ping a H4 desde H1. El ping debería realizarse correctamente.

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

* + - 1. Revisen los datos capturados en Wireshark. Examinen las direcciones IP y MAC a las que hicieron ping. Observen que la dirección MAC corresponde a la interfaz R1-eth1. Generen una lista con las direcciones IP y MAC de destino.

#### Pregunta:

Dirección IP:

Direcciones IP: 172.16.0.40.

Dirección MAC:

Dirección MAC: Esto estará asociado con la interfaz R1-eth1, que es el gateway predeterminado para los hosts H1, H2 y H3 en esta LAN.

* + - 1. En la ventana principal de la VM CyberOps, introduzcan **quit** para detener 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

* + - 1. Para borrar todos los procesos que utilizó Mininet, introduzcan el comando **sudo mn -c** en el cursor.

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

[sudo] contraseña para analyst:

\*\*\* 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.

Fin del documento