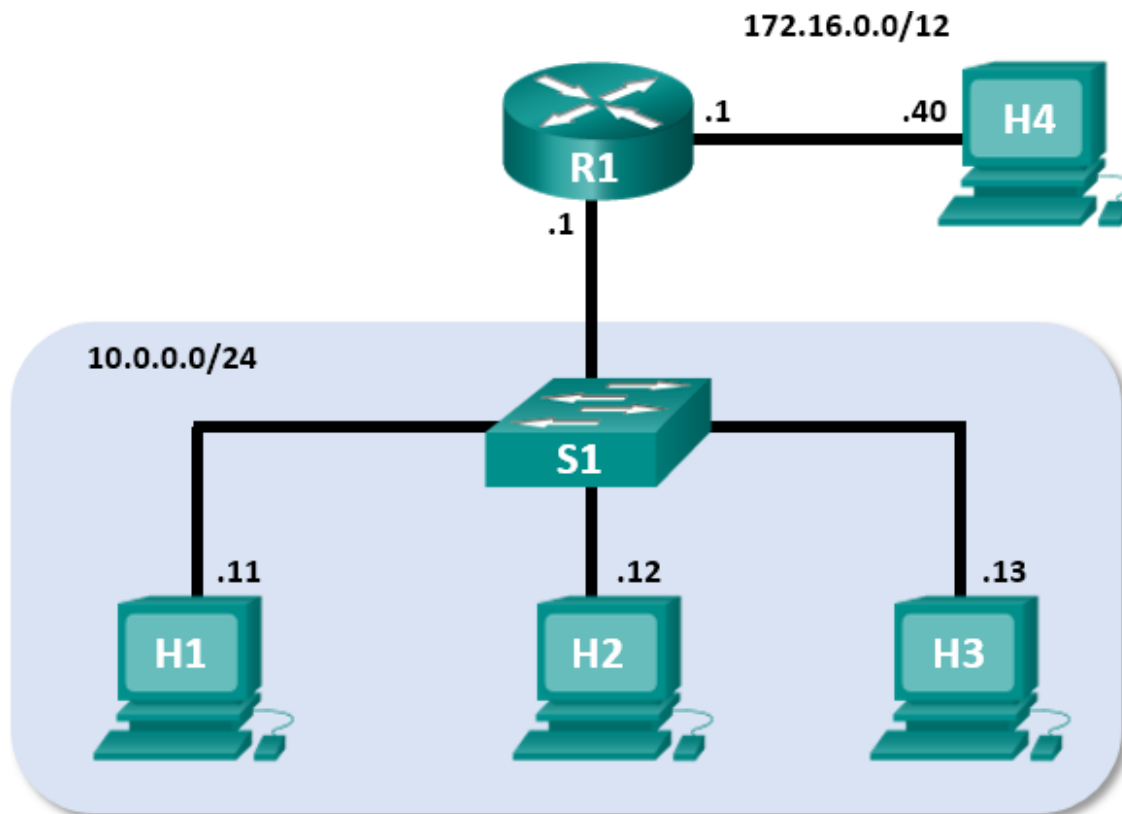


# Laboratorio - Introducción a Wireshark

## Topología de mininet



## Objetivos

**Parte 1: Instalar y verificar la topología de mininet**

**Parte 2: Captura y análisis de datos ICMP en Wireshark**

## Antecedentes / Escenario

La máquina virtual de CyberOps incluye un script de Python que, cuando lo ejecute, instalará y configurará los dispositivos que se muestran en la figura anterior. A continuación, tendrá acceso a cuatro hosts, un conmutador y un enrutador dentro de su única máquina virtual. Esto le permitirá simular una variedad de protocolos y servicios de red sin tener que configurar una red física de dispositivos. Por ejemplo, en este laboratorio usará el comando **ping** entre dos hosts en la topología de mininet y capturará esos pings con Wireshark.

Wireshark es un analizador de protocolo de software, o aplicación de "rastreo de paquetes", utilizado para la resolución de problemas de red, análisis, desarrollo de software y protocolos, y educación. A medida que los flujos de datos viajan a través de la red, el rastreador "captura" cada unidad de datos de protocolo (PDU) y puede decodificar y analizar su contenido de acuerdo con el RFC apropiada u otras especificaciones.

Wireshark es una herramienta útil para cualquiera que trabaje con redes para el análisis de datos y la resolución de problemas. Utilizará Wireshark para capturar paquetes de datos ICMP.

## Recursos requeridos

- Máquina virtual de CyberOps Workstation

## Instrucciones

### Parte 1: Instalar y verificar la topología de mininet

En esta parte, usará un script de Python para configurar la topología de mininet dentro de la máquina virtual de CyberOps. A continuación, registrará las direcciones IP y MAC para H1 y H2.

#### Paso 1: Verifique las direcciones de interfaz de su PC .

Inicie e inicie sesión en la estación de trabajo de CyberOps que haya instalado en un laboratorio anterior con las siguientes credenciales:

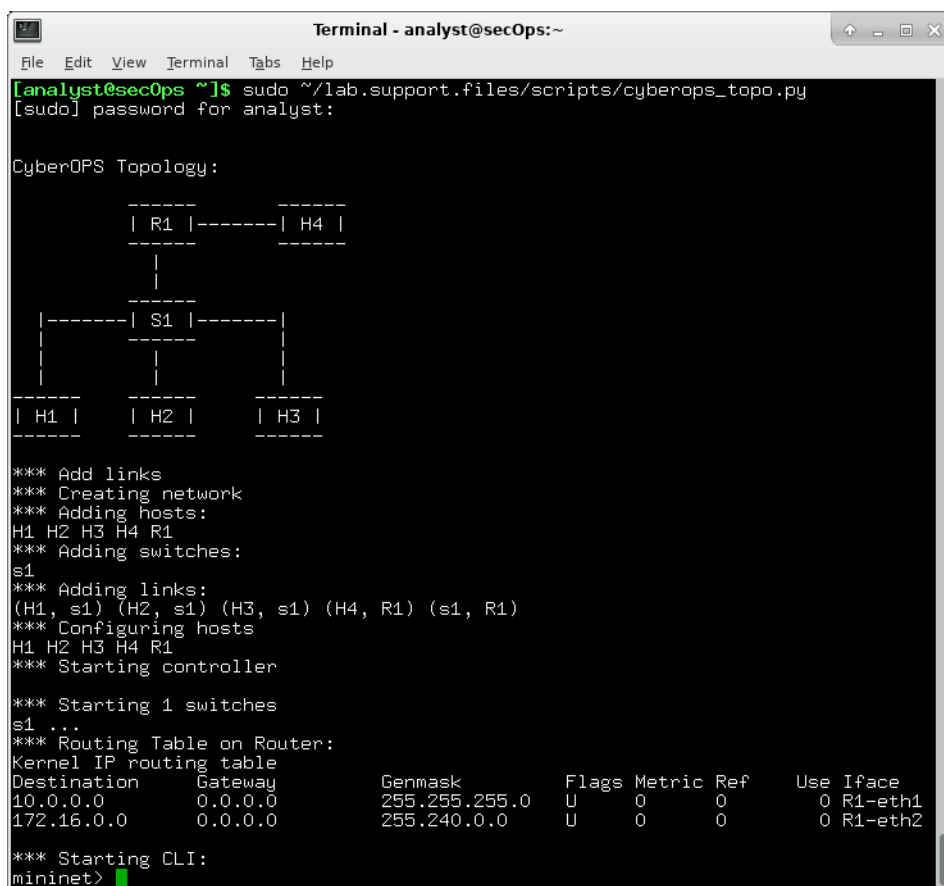
Nombre de usuario: **analista** Contraseña: **cyberops**

#### Paso 2: Ejecute el script de Python para instalar la topología de mininet.

Abra un emulador de terminal para iniciar Mininet e ingrese el siguiente comando en el indicador. Cuando se le solicite, ingrese **cyberops** como contraseña.

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

[sudo] Password para Analista:



```
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>
```

### Paso 3: Registre las direcciones IP y MAC para H1 y H2.

- a. En el indicador de mininet, inicie las ventanas de terminal en los hosts H1 y H2. Esto abrirá ventanas separadas para estos hosts. Cada host tendrá una configuración separada para la red que incluye direcciones IP y MAC únicas.

CLI inicial:

```
mininet> xterm H1
```

```
mininet> xterm H2
```

- b. En el indicador **Node: H1**, ingrese la dirección **IP** para verificar la dirección IPv4 y registrar la **dirección MAC**. Haga lo mismo para **Nodo: H2**. La dirección IPv4 y la dirección MAC se resaltan a continuación como referencia.

```
[root@secOps analista]# Dirección IP
```

```
< no se omite la salida>
```

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

```
link/ether ba:d 4:1d:7b:f3:61 brd ff :ff:ff:ff:ff link-netnsid 0 inet 10.0.0.11/24
```

```
brd 10.0.0.255 alcance global H1-eth0
```

```
valid_lft Forever preferred_lft Forever Inet6
```

```
fe80::B8D4:1DFF:FE7B:F361/64 Enlace de alcance
```

```
valid_lft Forever preferred_lft Forever
```

Interfaz de host	Dirección IP	Dirección MAC
H1-eth0		
H2-eth0		

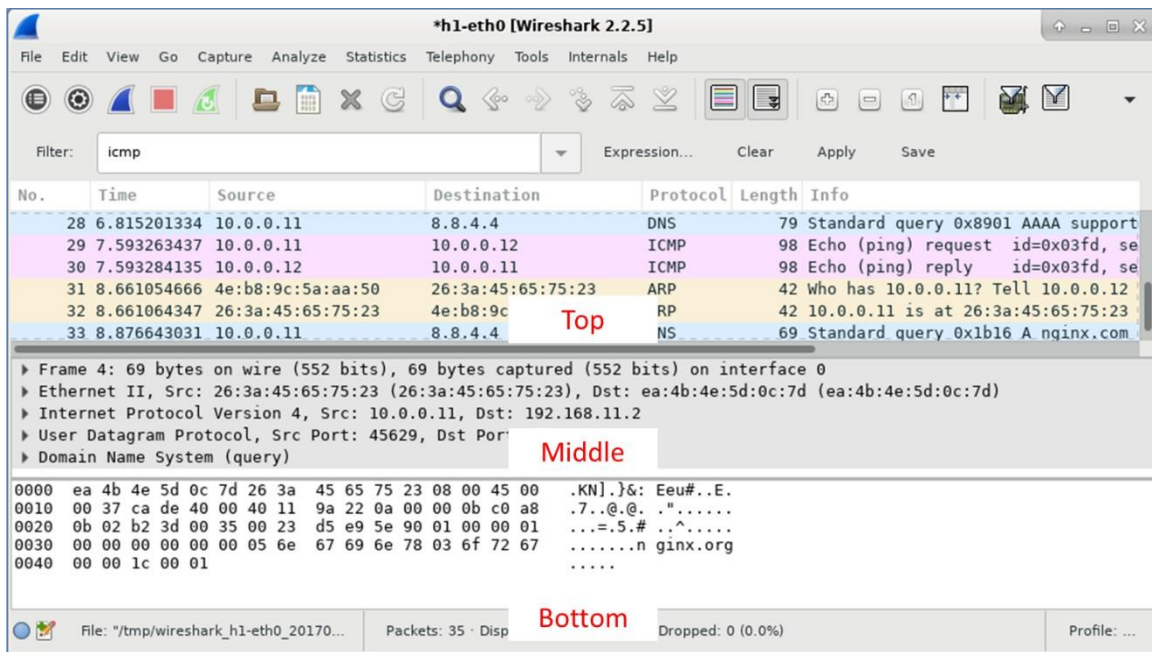
## Parte 2: Captura y análisis de datos ICMP en Wireshark

En esta parte, hará ping entre dos hosts en la Mininet y capturará solicitudes y respuestas ICMP en Wireshark. También buscará información específica dentro de las PDU capturadas. Este análisis debería ayudar a aclarar cómo se utilizan los encabezados de paquetes para transportar datos al destino.

### Paso 1: Examine los datos capturados en la misma LAN.

En este paso, examinará los datos generados por las solicitudes de ping del equipo de su miembro del equipo. Los datos de Wireshark se muestran en tres secciones:

- La sección superior muestra la lista de tramas PDU capturadas con un resumen de la información del paquete IP enumerado.
- La sección central muestra información de PDU para el fotograma seleccionado en la parte superior de la pantalla y separa un fotograma de PDU capturado por sus capas de protocolo.
- La sección inferior muestra los datos sin procesar de cada capa. Los datos sin procesar se muestran en forma hexadecimal y decimal.



- a. En **Nodo: H1**, ingrese **wireshark y** para iniciar Wireshark (La advertencia emergente no es importante para este laboratorio). Haga clic en **Aceptar** para continuar.

```
[root@secOps]# Wireshark &
```

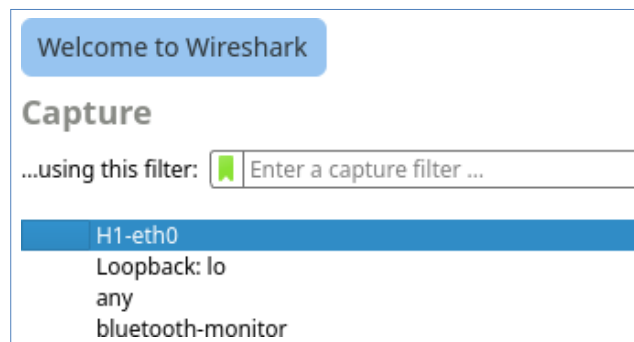
```
[1] 1552
```

```
[root@secOps ~]#
```

```
** (wireshark:1552): ADVERTENCIA **: No se pudo conectar al bus de accesibilidad: Error al conectarse al socket /tmp/dbus-f0dFz9baYA: Conexión rechazada
```

```
Gtk-Message: GtkDialog asignado sin un padre transitorio. Esto no se recomienda.
```

- b. En Wireshark window, bajo el encabezado **Captura**, seleccione la interfaz **H1-eth0**. Haga clic en **Inicio** para capturar el tráfico de datos.



- c. En **Nodo: H1**, presione la tecla Intro, si es necesario, para obtener un mensaje. Luego escriba **ping -c 5 10.0.0.12 a ping H2** cinco veces. La opción de comando **-c** especifica el recuento o número de pings. El **5** especifica que se deben enviar cinco pings. Todos los pings serán exitosos.

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

- d. Navegue hasta la ventana de Wireshark, haga clic en **Detener** para detener la captura de paquetes.
- e. Se puede aplicar un filtro para mostrar solo el tráfico interesado. Escriba **icmp** en el campo **Filtro** y haga clic en **Aplicar**.
- f. Si es necesario, haga clic en las primeras tramas PDU de solicitud ICMP en la sección superior de Wireshark. Observe que la columna Origen tiene la dirección IP de H1 y la columna Destino tiene la dirección IP de H2.

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) replv id=0x064e, seq=4/1024, ttl=64 (requ

- g. Con este marco de PDU aún seleccionado en la sección superior, navegue hasta la sección central. Haga clic en la flecha situada a la izquierda de la fila Ethernet II para ver las direcciones MAC de destino y de origen.

▶ 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

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

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

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

## Paso 2: Examine los datos capturados en la LAN remota.

Hará ping a los hosts remotos (hosts que no están en la LAN) y examinará los datos generados a partir de esos pings. A continuación, determinará qué es diferente acerca de estos datos de los datos examinados en la Parte 1.

- a. En el indicador de mininet, inicie las ventanas de terminal en los hosts H4 y R1.  

```
mininet> xterm H4
```

```
mininet> xterm R1
```
- b. En el indicador Node: **H4**, ingrese la dirección **IP** para verificar la dirección IPv4 y registrar la **dirección MAC**. Haga lo mismo para el **nodo: R1**.

[root@secOps analista]# **Dirección IP**

Interfaz de host	Dirección IP	Dirección MAC
H4-eth0		
R1-eth1		
R1-eth2		

- c. Inicie una nueva captura de Wireshark en H1 seleccionando **Capturar > Iniciar**. También puede hacer clic en el botón **Inicio** o escribir **Ctrl-E**. Haga clic en **Continuar sin guardar** para iniciar una nueva captura.
- d. H4 es un servidor remoto simulado. Ping H4 desde H1. El ping debería ser correcto.

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

- e. Revise los datos capturados en Wireshark. Examine las direcciones IP y MAC a las que ha hecho ping. Observe que la dirección MAC es para la interfaz R1-eth1. Enumere las direcciones IP y MAC de destino .

Dirección IP:

Dirección MAC :

- f. En la ventana principal de CyberOps VM, escriba **quit** to stop Mininet.

Mininet> **salir**

Detener 0 controladores

Detener 4 términos

Detener 5 enlaces

.....

Parada 1 interruptores s1

Detener 5 hosts H1 H2 H3

H4 R1

Hecho

- g. Para limpiar todos los procesos utilizados por Mininet, ingrese el comando **sudo mn -c** en el símbolo del sistema.

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

[sudo] Contraseña para el analista:

Eliminación del exceso de controladores/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"

Eliminar basura de /tmp

rm -f /tmp/vconn\* /tmp/vlogs\* /tmp/\*.out /tmp/\*.log

Eliminación de antiguos túneles X11

Eliminación de rutas de datos de kernel excedentes

PS AX | egrep -o 'dp[0-9]+' | sed 's/dp/nt:/'

Eliminación de rutas de datos de OVS

ovs-vsctl --timeout=1 list-br

ovs-vsctl --timeout=1 list-br

Eliminación de todos los enlaces del patrón foo-ethX

Enlace IP Mostrar | egrep -o '([\_.: :alnum:])+eth[:d igit:])+)' ip link show

Matar procesos de nodo de mininet obsoletos

pkill -9 -f mininet:

Cierre de túneles obsoletos pkill -9 -f

Túnel = Ethernet pkill -9 -f . ssh/mn

rm -f ~/. ssh/mn/\*

Limpieza completa.