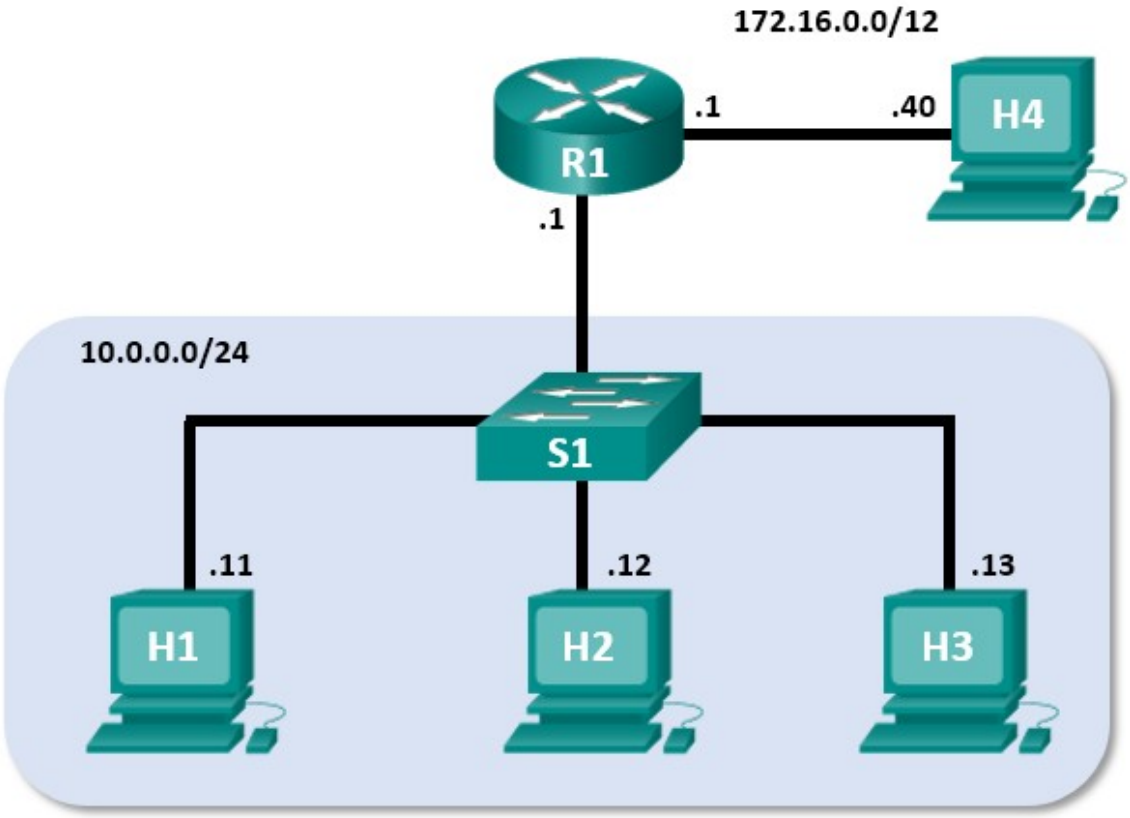
Laboratório - Introdução ao Wireshark

Topologia do Mininet



Objetivos

Parte 1: Instalar e verificar a topologia do Mininet

Parte 2: capturar e analisar dados ICMP no Wireshark

Histórico/Cenário

A VM CyberOps inclui um script Python que, quando você executá-lo, configura os dispositivos mostrados na figura acima. Você terá acesso a quatro hosts, um switch e um roteador dentro de uma VM. Isso permitirá que você simule uma variedade de protocolos e serviços de rede sem ter que configurar uma rede física de dispositivos. Por exemplo, neste laboratório você usará o comando **ping** entre dois hosts na Topologia Mininet e capturará esses pings com Wireshark.

O Wireshark é um software analisador de protocolo, ou uma aplicação "packet sniffer", usado para solução de problemas de rede, análise, desenvolvimento de software e protocolo, e educação. Conforme os fluxos de dados trafegam pela rede, o farejador "captura" cada unidade de dados de protocolo (PDU) e pode decodificar e analisar seu conteúdo de acordo com a RFC apropriada ou outras especificações.

O Wireshark é uma ferramenta útil para qualquer pessoa que trabalhe com redes para análise de dados e solução de problemas. Você usará o Wireshark para capturar pacotes de dados ICMP.

Recursos necessários

- Máquina Virtual CyberOps Workstation

Instruções

Parte 1: Instalar e verificar a topologia do mininet

Nesta parte, você usará um script Python para configurar a Topologia Mininet dentro da VM CyberOps. Em seguida, você registra os endereços IP e MAC para H1 e H2.

Etapa 1: Verifique os endereços de interface do seu PC.

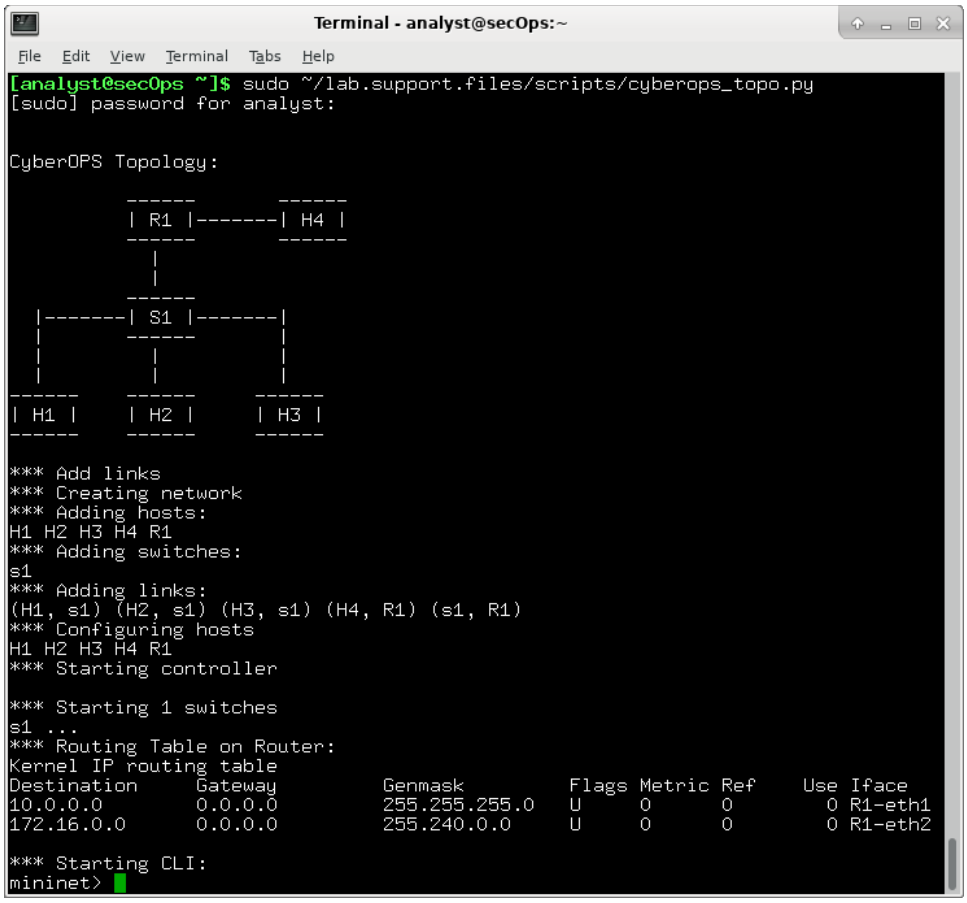
Inicie e faça login na CyberOps Workstation que você instalou em um laboratório anterior usando as seguintes credenciais:

Nome de usuário: **analyst** Senha: **cyberops**

Etapa 2: Execute o script Python para instalar a Topologia do Mininet.

Abra um emulador de terminal para iniciar o Mininet e digite o seguinte comando no prompt. Quando solicitado, digite **cyberops** como a senha.

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



Etapa 3: Registre endereços IP e MAC para H1 e H2.

a. No prompt da mininet, inicie as janelas do terminal nos hosts H1 e H2. Isso abrirá janelas separadas para esses hosts. Cada host terá uma configuração separada para a rede, incluindo endereços IP e MAC exclusivos.

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

b. No prompt em **Node: H1**, digite o **endereço IP** para verificar o endereço IPv4 e registrar o endereço MAC. Faça o mesmo para **Node: H2**. O endereço IPv4 e o endereço MAC são destacados abaixo para referência.

```
[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/éter ba:d 4:1 d:7b:f 3:61 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.0.11/24 brd 10.0.0.255 escopo global H1-eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::b8d4:1dff:fe7b:f361/64 scope link
       valid_lft forever preferred_lft forever
```

Interface de host	Endereço IP	Endereço MAC
H1-eth0		
H2-eth0		

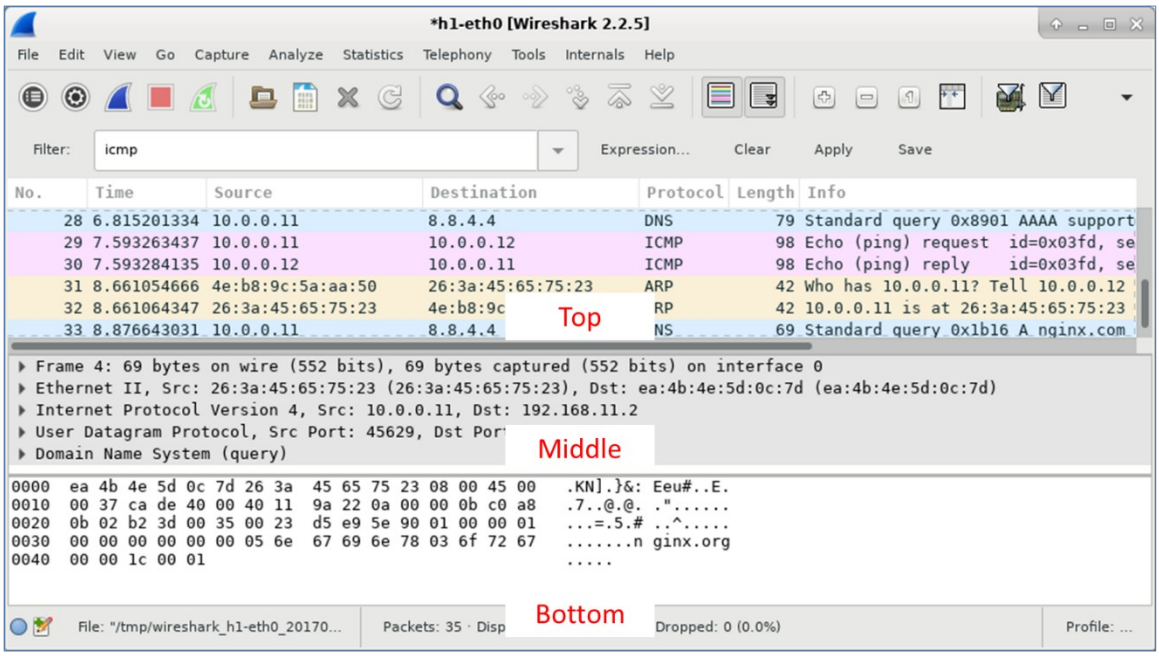
Parte 2: Capture e analise dados ICMP no Wireshark

Nesta parte, você fará ping entre dois hosts no Mininet e capturará solicitações e respostas ICMP no Wireshark. Você também examinará as PDUs capturadas para obter informações específicas. Essa análise deve ajudar a esclarecer como os cabeçalhos dos pacotes são usados para transportar dados ao destino.

Etapa 1: Examine os dados capturados na mesma LAN.

Nesta etapa, você examinará os dados gerados pelas solicitações de ping do PC de seu membro da equipe. Os dados do Wireshark são exibidos em três seções:

- A seção superior exibe a lista de quadros de PDU capturados com um resumo das informações do pacote IP listadas.
- A seção média lista as informações de PDU do quadro selecionado na parte superior da tela e separa um quadro de PDU capturado pelas suas camadas de protocolo.
- A seção inferior mostra os dados brutos de cada camada. Os dados são exibidos em formato hexadecimal e decimal.

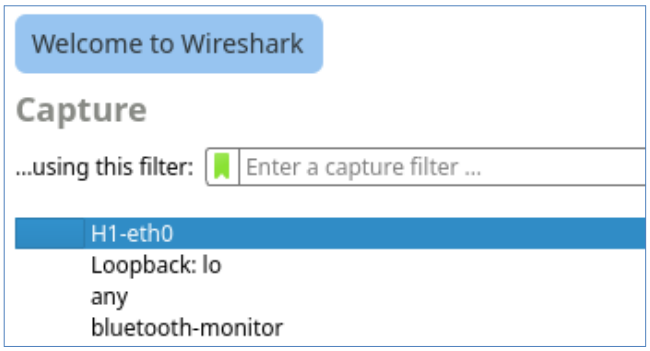


a. No **nó: H1**, digite **wireshark &** para iniciar o Wireshark (O aviso pop-up não é importante para este laboratório). Clique em **OK** para continuar.

```
[root @secOps] # wireshark-gtk &
[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. Isto é
desencorajado.
```

b. Na janela Wireshark, sob o título **Capture**, selecione a interface **H1-eth0**. Clique em **Start** para capturar o tráfego de dados.



c. Em **Node: H1**, pressione a tecla Enter, se necessário, para obter um prompt. Em seguida, digite **ping -c 5 10.0.0.12** para realizar o ping H2 cinco vezes. A opção de comando **-c** especifica a contagem ou o número de pings. O **5** especifica que cinco pings devem ser enviados. Os pings serão todos bem-sucedidos.

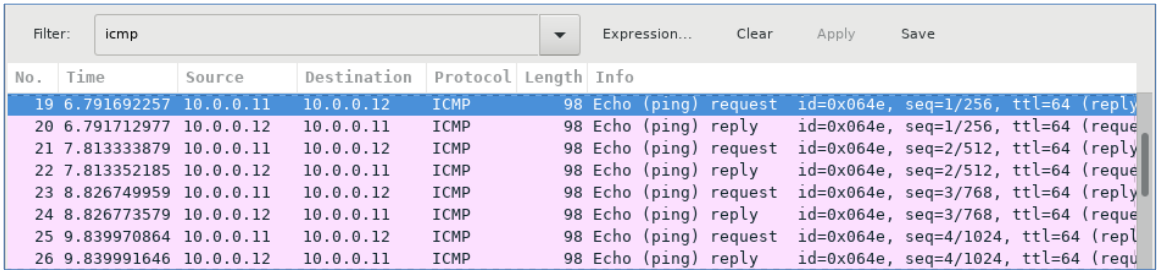
```
[root @secOps analista] # ping -c 5 10.0.0.12
```

d. Navegue até a janela Wireshark, clique em **Stop** para interromper a captura de pacotes.

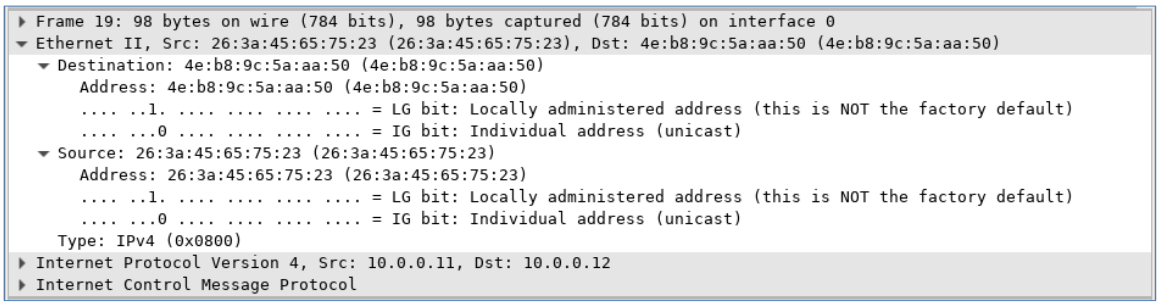
e. Um filtro pode ser aplicado para exibir apenas o tráfego interessado.

Digite **icmp** no campo **Filter** e clique em **Apply**.

f. Se necessário, clique nos primeiros quadros de PDU de solicitação de ICMP na seção superior do Wireshark. Observe que a coluna Origem tem o endereço IP do H1 e a coluna Destino tem o endereço IP do H2.



g. Com esse quadro de PDU ainda selecionado na seção superior, vá até a seção média. Clique na seta à esquerda da linha Ethernet II para ver os endereços MAC de origem e destino.



O endereço MAC de origem corresponde à interface de H1?

O endereço MAC de destino no Wireshark corresponde ao endereço MAC de H2?

Nota: No exemplo anterior de uma solicitação ICMP capturada, os dados ICMP são encapsulados dentro de uma PDU de pacote IPv4 (cabeçalho IPv4), que é então encapsulado em uma PDU de quadro Ethernet II (cabeçalho Ethernet II) para transmissão na LAN.

Etapa 2: Examine os dados capturados na LAN remota.

Você executará ping em hosts remotos (hosts que não estão na LAN) e examinará os dados gerados a partir desses pings. Você determinará o que há de diferente nesses dados a partir dos dados pesquisados na parte 1.

a. No prompt da mininet, inicie as janelas do terminal nos hosts H4 e R1.

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

b. No prompt em **Node: H4**, digite o **ip address** para verificar o endereço IPv4 e registrar o endereço MAC. Faça o mesmo para o **node: R1**.

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

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

c. Inicie uma nova captura Wireshark em H1 selecionando **Capture > Start**. Você também pode clicar no botão **Start** ou digitar **Ctrl-E** Clique em **Continue without Saving** para iniciar uma nova captura.

d. H4 é um servidor remoto simulado. Ping H4 de H1. O ping deve obter êxito.

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

e. Revise os dados capturados no Wireshark. Examine os endereços IP e MAC que você fez ping. Observe que o endereço MAC é para a interface R1-eth1. Liste os endereços IP e MAC de destino.

Endereço IP:

Endereço MAC :

f. Na janela principal da VM do CyberOps, digite **quit** para parar o Mininet.

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

```
*** Stopping 4 terms
*** Stopping 5 links
.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Stopping 5 hosts
```

g. Para limpar todos os processos que foram usados pela Mininet, digite o comando **sudo mn -c** no prompt.

```
[analyst @secOps ~] $ sudo mn -c
[sudo] password for 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 excess controllers/ofprotocols/ofdatapaths/pings/noxes
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 OVS datapaths
ip link show | egrep -o '([-_.[:alnum:]] +-eth [[:dígito:]] +) '
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Túnel = Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```