

Informática Forense y Auditoría

Grado en Ingeniería Informática del Software

Escuela de Ingeniería Informática de Oviedo

Práctica 5 bis: Ejercicios de Práctica Forense V

Introducción

Los protocolos de red son un elemento que aparece con asiduidad en cualquier escenario forense. El investigador forense debe, en ocasiones, analizar el tráfico de red intercambiado entre los sistemas implicados para detectar si se ha producido un incidente (intrusión, descarga de malware, etc.).

Objetivos

- Familiarizar al alumno con el software de análisis de protocolos Wireshark.
- Analizar, utilizando Wireshark, la negociación (handshake) para establecer una sesión TCP.
- Utilizar Wireshark para examinar los intercambios de consulta y respuesta con un servidor DNS.
- Capturar y analizar tráfico HTTP y HTTPS.
- Capturar y analizar los campos de encabezado del protocolo TCP para las transferencias de archivos mediante el protocolo FTP.
- Capturar y analizar el tráfico generado por sesiones de Telnet y SSH y comprender la importancia del cifrado.
- Examinar cómo es un ataque a un sitio web mediante inyección de código SQL.
- Examinar el tráfico capturado en un archivo PCAP y extraer del mismo un archivo “malicioso”.

Instrucciones comunes a todos los ejercicios

- Deberá justificar la realización de los siguientes ejercicios con capturas de pantalla donde quede patente la autoría de los mismos, así como los comandos empleados en su resolución y aquellas explicaciones que considere oportunas.

Forensia de redes con Wireshark

1. En este ejercicio veremos una introducción a Wireshark, herramienta que manejaremos en diferentes ejercicios de práctica forense.

Trabajaremos con la topología MiniNet la cual se muestra a continuación:

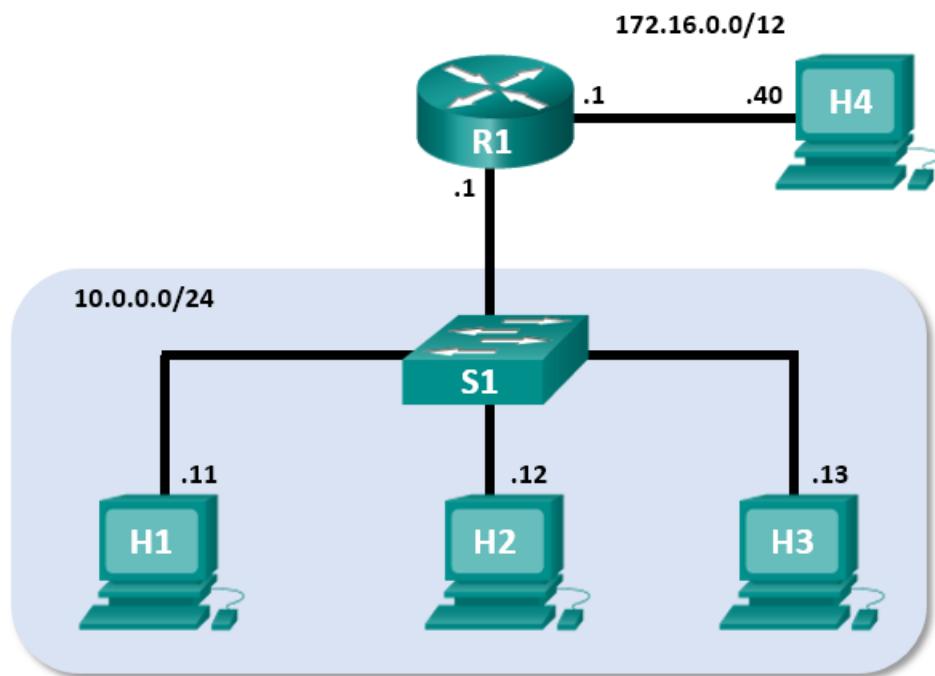


Figura 1. Topología Mininet

Objetivos del ejercicio:

- Instalar y verificar la topología MiniNet.
- Capturar y analizar paquetes del protocolo ICMP en Wireshark

Preparación del entorno de pruebas:

Mininet es un emulador de red que utiliza virtualización ligera para crear redes virtuales para la creación rápida de prototipos de diseños de redes definidas por software (SDN) utilizando OpenFlow.

Descarga del campus virtual un script denominado **Recursos Prácticas-> Práctica 5-> minired.py**.

Transfiérelo a la máquina virtual de CAINE (puedes utilizar el Filezilla, para ello activando previamente en dicha máquina el servidor ssh). Para activar el servicio ssh en la máquina virtual de CAINE ejecuta el siguiente comando:

```
sudo systemctl start ssh.service
```

Dale permisos de ejecución al script (**chmod ugo+x minired.py**)

En la máquina virtual de CAINE viene preinstalado el intérprete de Python en su versión 2.7.15+. Para poder ejecutar el script necesitamos previamente instalar el paquete **mininet** en CAINE. Para ello ejecuta los siguientes comandos:

```
sudo apt update
```

```
sudo apt -y install mininet
```

Vete a la carpeta donde has descargado el script y ejecuta el comando **minired.py** con privilegios de administrador.

```
marco@marco-VirtualBox: ~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ ls -la
total 12
drwxrwxr-x  2 marco marco 4096 giu 14 13:29 .
drwxr-xr-x 34 marco marco 4096 giu 14 13:29 ..
-rwxrwxr-x  1 marco marco 3669 giu 13 13:30 minired.py
marco@marco-VirtualBox:~/practicas_wireshark$ sudo ./minired.py
```

```
marco@marco-VirtualBox: ~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda

Topology:

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

*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
```

```

marco@marco-VirtualBox: ~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda

-----
*** 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:
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask     Indic Métric Ref   Uso Interfaz
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> 

```

Cuando ejecutes el script, instalará y configurará los dispositivos que se muestra en la Figura 1. Luego tendrá acceso a cuatro hosts, un switch y un enrutador dentro de su única máquina virtual. Esto te permitirá simular una variedad de protocolos y servicios de red sin tener que configurar una red física de dispositivos. Por ejemplo, en este ejercicio utilizarás el comando ping entre dos hosts en la topología de Mininet y capturarás esos ping con Wireshark.

Wireshark es un analizador de protocolos de software, o una aplicación de "sniffer de paquetes", que se utiliza para la solución de problemas de red, el análisis de redes, el desarrollo de software de protocolos y la educación. A medida que los flujos de datos viajan por la red, el sniffer "captura" cada unidad de datos de protocolo (PDU) y puede decodificar y analizar su contenido de acuerdo con el RFC apropiado u otras especificaciones.

Wireshark es una herramienta útil para cualquier persona que trabaje con redes para el análisis de datos y la resolución de problemas. Utilizará Wireshark para capturar paquetes de datos ICMP (que son los que genera la utilidad de **ping**).

Parte 1:

- En el prompt de mininet, inicia 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.

```
*** Routing Table on Router:
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref    Uso Interfaz
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> 
```



- b) En el prompt del Nodo: H1, escribe **ip a** para verificar la dirección IPv4 y registrar la dirección MAC. Haga lo mismo para el Nodo: H2. La dirección IPv4 y la dirección MAC se destacan a continuación como referencia.

```
"Node: H1" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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
    inetc6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: H1-eth0@if21: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 3e:a8:c7:77:ae:4d 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
    inetc6 fe80::c0e7:82ff:fe18:13cc/64 scope link
        valid_lft forever preferred_lft forever
root@marco-VirtualBox:~/practicas_wireshark# 
```



```
"Node: H2" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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
    inetc6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: H2-eth0@if22: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether c2:c7:82:18:13:cc brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.0.12/24 brd 10.0.0.255 scope global H2-eth0
        valid_lft forever preferred_lft forever
    inetc6 fe80::c0e7:82ff:fe18:13cc/64 scope link
        valid_lft forever preferred_lft forever
root@marco-VirtualBox:~/practicas_wireshark# 
```

- c) Rellena los datos siguientes de acuerdo a la información que te proporciona el comando anterior ejecutado en cada una de las terminales correspondientes a los Nodos H1 y H2 de tu máquina:

Host-Interfaz	Dirección IPv4	Dirección MAC
H1-eth0		
H2-eth0		

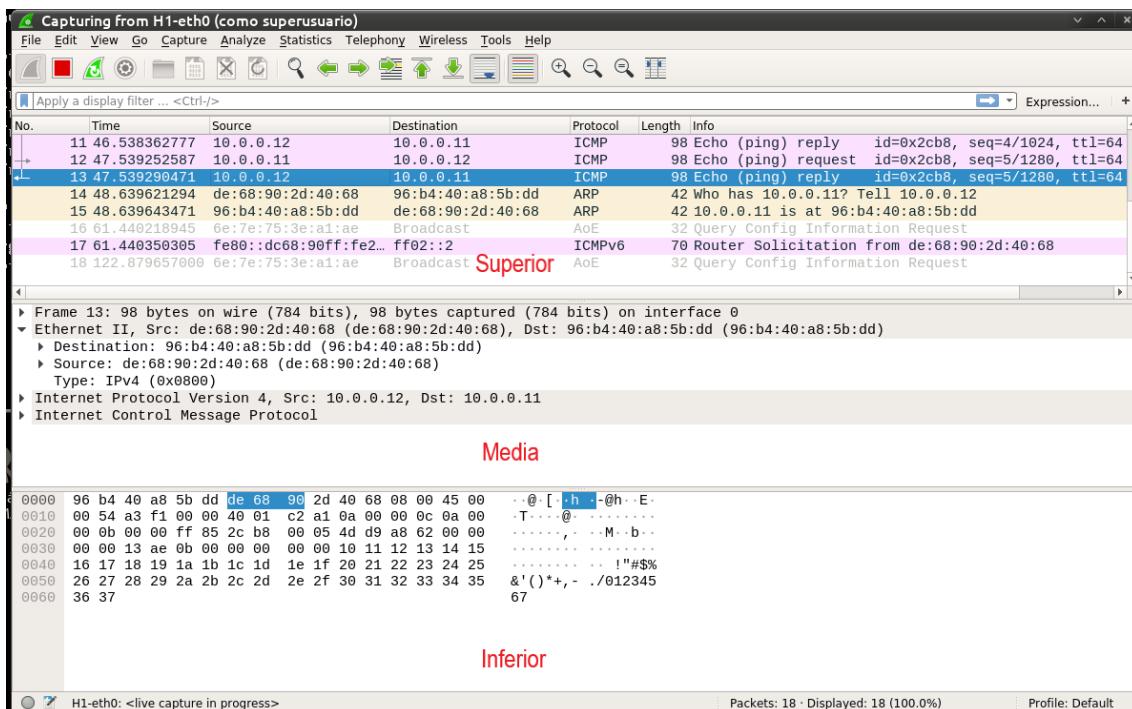
Parte 2:

En esta parte, hará ping entre dos hosts en 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 los paquetes para transportar datos al destino.

Paso 1: Examinar los datos capturados de la misma LAN

En este paso, examinará los datos generados por las solicitudes de ping de los PCs emulados en Mininet. Los datos de Wireshark se muestran en tres secciones:

- La sección superior muestra la lista de tramas de PDU capturadas con un resumen de la información del paquete IP enumerada.
- La sección central enumera la información de PDU para la fila seleccionada en la parte superior de la pantalla y separa la PDU capturada por sus capas de protocolo.
- La sección inferior muestra los datos sin procesar de cada capa. Los datos sin procesar se muestran en formato hexadecimal y decimal.

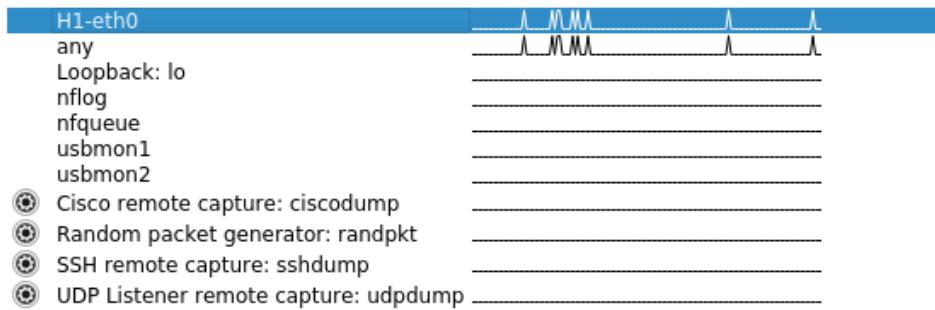


- En el nodo: H1, ingrese **wireshark &** para iniciar Wireshark (la advertencia emergente no es importante para este laboratorio). Haga clic en Aceptar para continuar.
- En la ventana Wireshark, bajo el encabezado Captura, seleccione la interfaz H1-eth0. Haga clic en Iniciar para capturar el tráfico de datos.

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ...



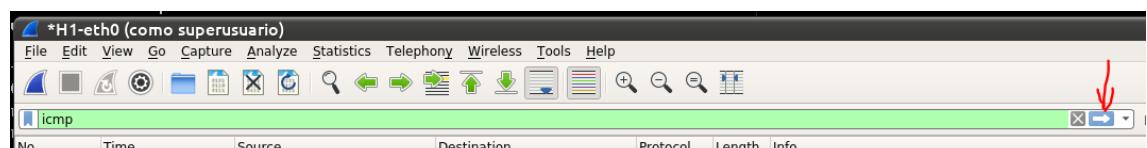
- f) En el Nodo: H1, presione la tecla Intro, si es necesario, para obtener el prompt. Luego escriba **ping -c 5 10.0.0.12** para hacer ping a H2 cinco veces. La opción de comando **-c** especifica el recuento o el número de pings. El 5 especifica que se deben enviar cinco pings. Todos los pings serán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

```
"Node: H1" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark# wireshark &
[1] 11336
root@marco-VirtualBox:~/practicas_wireshark# QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

root@marco-VirtualBox:~/practicas_wireshark# ping -c 5 10.0.0.12
PING 10.0.0.12 (10.0.0.12) 56(84) bytes of data.
64 bytes from 10.0.0.12: icmp_seq=1 ttl=64 time=0.502 ms
64 bytes from 10.0.0.12: icmp_seq=2 ttl=64 time=0.145 ms
64 bytes from 10.0.0.12: icmp_seq=3 ttl=64 time=0.126 ms
64 bytes from 10.0.0.12: icmp_seq=4 ttl=64 time=0.125 ms
64 bytes from 10.0.0.12: icmp_seq=5 ttl=64 time=0.067 ms

--- 10.0.0.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.067/0.193/0.502/0.156 ms
root@marco-VirtualBox:~/practicas_wireshark#
```

- g) Vaya a la ventana de Wireshark, haga clic en Detener para detener la captura de paquetes.
h) Se pueden aplicar filtros para mostrar solo el tráfico de un determinado protocolo que nos interesa. Teclea **icmp** en el campo **Filter** y aplícalo.



- i) Si es necesario, haga clic en las primeras filas de 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.

No.	Time	Source	Destination	Protocol	Length	Info
5	43.534248637	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0xcb8, seq=1/256, ttl=
6	44.534393407	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0xcb8, seq=2/512, ttl=
7	44.534473692	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0xcb8, seq=2/512, ttl=
8	45.536083766	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0xcb8, seq=3/768, ttl=
9	45.536152175	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0xcb8, seq=3/768, ttl=
10	46.538294697	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0xcb8, seq=4/1024, ttl=
11	46.538362777	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0xcb8, seq=4/1024, ttl=
12	47.539252587	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0xcb8, seq=5/1280, ttl=
13	47.539290471	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0xcb8, seq=5/1280, ttl=

- j) Con esta fila de PDU aún seleccionada en la sección superior, navegue a la sección central. Haga clic en la flecha a la izquierda de la fila Ethernet II para ver las direcciones MAC de origen y destino.

```

Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: de:68:90:2d:40:68 (de:68:90:2d:40:68), Dst: 96:b4:40:a8:5b:dd (96:b4:40:a8:5b:dd)
  ▶ Destination: 96:b4:40:a8:5b:dd (96:b4:40:a8:5b:dd)
  ▶ Source: de:68:90:2d:40:68 (de:68:90:2d:40:68)
    Type: IPv4 (0x0800)
  ▶ Internet Protocol Version 4, Src: 10.0.0.12, Dst: 10.0.0.11
  ▶ Internet Control Message Protocol

```

- k) ¿La dirección MAC de origen coincide con la interfaz de H1? Conteste a la vista de los resultados que proporciona su propio sistema.
l) ¿La dirección MAC de destino en Wireshark coincide con la dirección MAC de H2? Conteste a la vista de los resultados que proporciona su propio sistema.

Paso 2: Examinar los datos capturados en la LAN remota.

En este paso, harás ping a hosts remotos (hosts que no están en la misma LAN) y examinarás los datos generados a partir de esos pings. Luego, determinarás en qué se diferencian estos datos de los datos examinados en la Parte 1.

- m) En el prompt de mininet, inicie las ventanas de terminal en los hosts H4 y R1.

```

mininet> xterm H4
mininet> xterm R1
mininet> 

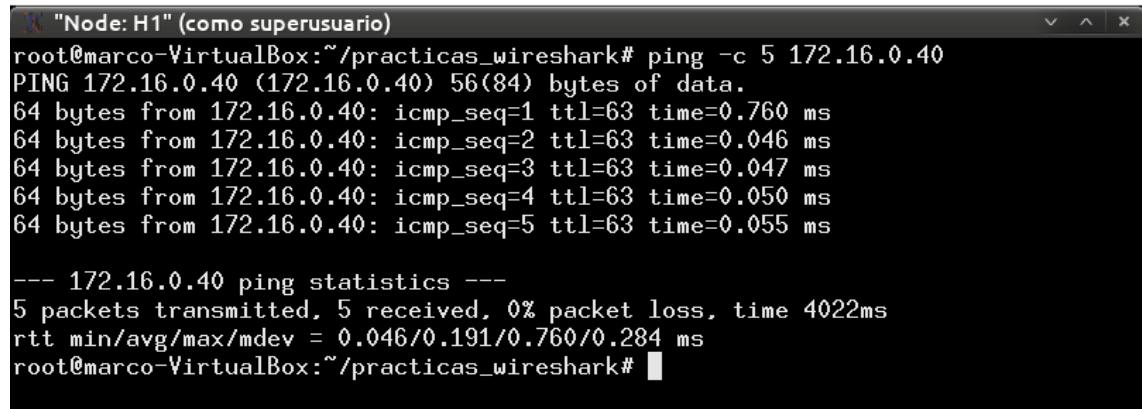
```

- n) En el prompt del Nodo: H4, ejecute **ip a** para verificar la dirección IPv4 y registrar la dirección MAC. Haga lo mismo para el Nodo: R1.

Host-Interfaz	Dirección IPv4	Dirección MAC
H4-eth0		
R1-eth1		
R1-eth2		

- o) Inicia una nueva captura de Wireshark en H1 seleccionando **Capturar> Iniciar**. También puedes hacer clic en el botón Inicio o pulsar Ctrl-E. Haz clic en Continuar sin guardar para iniciar una nueva captura.

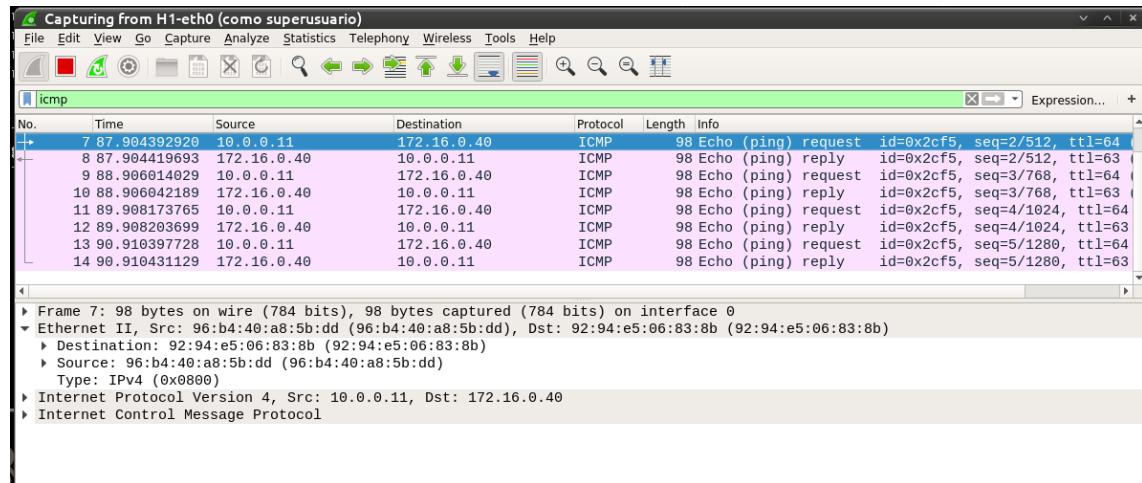
- p) H4 es un servidor remoto simulado. Haz ping a H4 desde H1. El ping debería tener éxito.



```
"Node: H1" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark# ping -c 5 172.16.0.40
PING 172.16.0.40 (172.16.0.40) 56(84) bytes of data.
64 bytes from 172.16.0.40: icmp_seq=1 ttl=63 time=0.760 ms
64 bytes from 172.16.0.40: icmp_seq=2 ttl=63 time=0.046 ms
64 bytes from 172.16.0.40: icmp_seq=3 ttl=63 time=0.047 ms
64 bytes from 172.16.0.40: icmp_seq=4 ttl=63 time=0.050 ms
64 bytes from 172.16.0.40: icmp_seq=5 ttl=63 time=0.055 ms

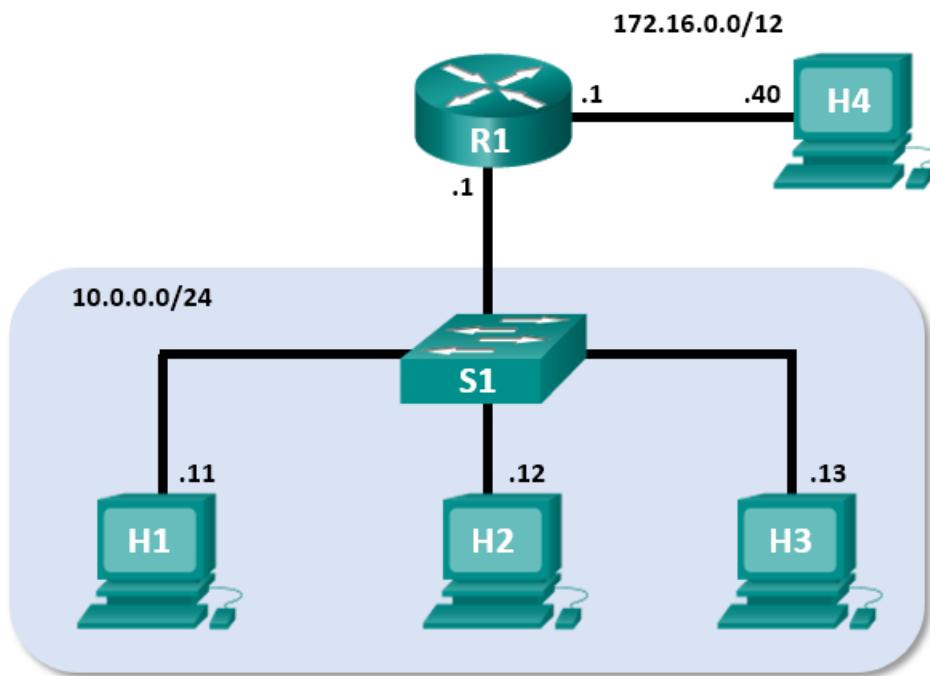
--- 172.16.0.40 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4022ms
rtt min/avg/max/mdev = 0.046/0.191/0.760/0.284 ms
root@marco-VirtualBox:~/practicas_wireshark#
```

- q) Revisa los datos capturados en Wireshark. Examina las direcciones IP y MAC a las que hiciste ping. Ten en cuenta que la dirección MAC es para la interfaz R1-eth1. Enumera las direcciones IP y MAC de destino.



- r) En la ventana desde la que lanzaste el script de minired.py, escribe quit para detener Mininet.
- s) Para limpiar todos los procesos que utilizaste en Mininet, escribe el comando sudo mn -c en el prompt.
2. En esta ejercicio usaremos Wireshark para observar el protocolo de enlace TCP de 3 pasos (Handshake). Cuando una aplicación, como HTTP o el protocolo de transferencia de archivos (FTP), se inicia en un host, TCP utiliza la negociación en tres pasos para establecer una sesión de TCP confiable entre los dos hosts. Por ejemplo, cuando en un PC utilizas un navegador web para navegar por Internet, se inicia una negociación en tres pasos y se establece una sesión entre el host del PC y el servidor web. Un PC puede tener varias sesiones de TCP activas simultáneas con varios sitios web.

Trabajaremos con la topología MiniNet la cual se muestra a continuación:



Objetivos de la práctica:

- Utilizar Wireshark para capturar y examinar los paquetes generados entre el navegador del PC utilizando el protocolo de transferencia de hipertexto (HTTP) y un servidor web.

Preparación del entorno de pruebas:

Descarga del campus virtual un script denominado **Recursos Prácticas-> Práctica 5->nginx_start.sh**.

Transfiérelo a la máquina virtual de CAINE (puedes utilizar el Filezilla, para ello activando previamente en dicha máquina el servidor ssh).

Dale permisos de ejecución al script (**chmod ugo+x nginx_start.sh**)

La máquina virtual de CAINE no trae preinstalado el servidor HTTP **nginx**. Desde una ventana de terminal lance el siguiente comando para instalar el servidor HTTP **nginx**.

```

marco@marco-VirtualBox: ~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ sudo apt install nginx

```

Vamos a crear la red de pruebas Mininet tal como hicimos en el ejercicio anterior. Para ello ejecuta el comando **minired.py** con privilegios de administrador.

```
marco@marco-VirtualBox:~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ ls -la
total 12
drwxrwxr-x  2 marco marco 4096 giu 14 13:29 .
drwxr-xr-x 34 marco marco 4096 giu 14 13:29 ..
-rwxrwxr-x  1 marco marco 3669 giu 13 13:30 minired.py
marco@marco-VirtualBox:~/practicas_wireshark$ sudo ./minired.py
```

```
marco@marco-VirtualBox:~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda

Topology:

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

*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:

```

marco@marco-VirtualBox: ~/practicas_wireshark
Archivo Editar Ver Buscar Terminal Ayuda
-----
*** 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:
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref   Uso Interfaz
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> 

```

Parte 1:

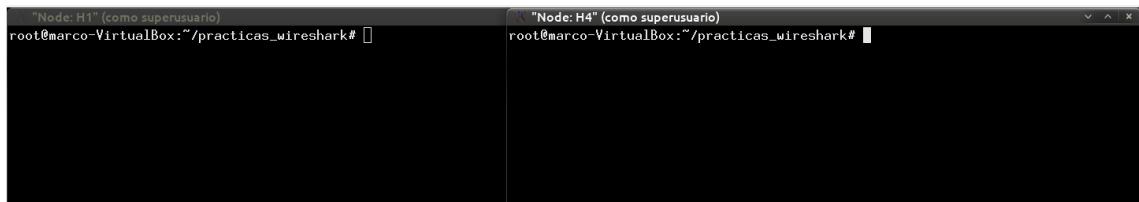
- En el prompt de mininet, inicie las ventanas de terminal en los hosts H1 y H4. Esto abrirá ventanas separadas para estos hosts. Cada host tendrá una configuración separada para la red que incluye direcciones IP y MAC únicas.

```

*** Routing Table on Router:
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref   Uso Interfaz
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 H4
mininet> 

```



- Inicia el servidor web en H4 ejecutando el script **nginx_start.sh**.

```
"Node: H4" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark# ./nginx_start.sh
root@marco-VirtualBox:~/practicas_wireshark#
```

- c) Por motivos de seguridad, no está admitido el uso de Firefox para la cuenta root. En el host H1, utiliza el comando su ("switch user") para cambiar del usuario root a la cuenta de usuario con la que hayas iniciado sesión en la máquina de CAINE.

```
"Node: H1" (como superusuario)
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark#
root@marco-VirtualBox:~/practicas_wireshark# su marco
marco@marco-VirtualBox:~/practicas_wireshark$
```

- d) Inicia el navegador web firefox en H1. En unos instantes aparecerá la ventana del navegador firefox.

```
"Node: H1" (como superusuario)
marco@marco-VirtualBox:~/practicas_wireshark$ firefox 2> /dev/null &
[1] 14198
marco@marco-VirtualBox:~/practicas_wireshark$
```

- e) Despues de que se abra la ventana de Firefox, inicia una sesión de **tcpdump** (investiga las posibilidades de este comando con el man) en el terminal Nodo: H1 y envía la salida a un archivo de nombre **captura.pcap** en la carpeta que elija. Con la opción -v pueden ver el progreso. Esta captura se detendrá después de capturar 50 paquetes, porque está configurada con la opción -c 50.

```
"Node: H1" (como superusuario)
marco@marco-VirtualBox:~/practicas_wireshark$ firefox 2> /dev/null &
[1] 14198
marco@marco-VirtualBox:~/practicas_wireshark$ sudo tcpdump -i H1-eth0 -v -c 50
-w /home/marco/practicas_wireshark/captura.pcap
```

- f) Despues de que se inicie tcpdump, diríjase rápidamente a 172.16.0.40 en el navegador web Firefox.

```
"Node: H1" (como superusuario)
marco@marco-VirtualBox:~/practicas_wireshark$ firefox 2> /dev/null &
[1] 14198
marco@marco-VirtualBox:~/practicas_wireshark$ sudo tcpdump -i H1-eth0 -v -c 50
-w /home/marco/practicas_wireshark/captura.pcap
tcpdump: listening on H1-eth0, link-type EN10MB (Ethernet), capture size 262144
bytes
50 packets captured
52 packets received by filter
Welcome to nginx! - Mozilla Firefox
Welcome to nginx!    x  +
← → C  ↻  172.16.0.40
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and
Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.
```

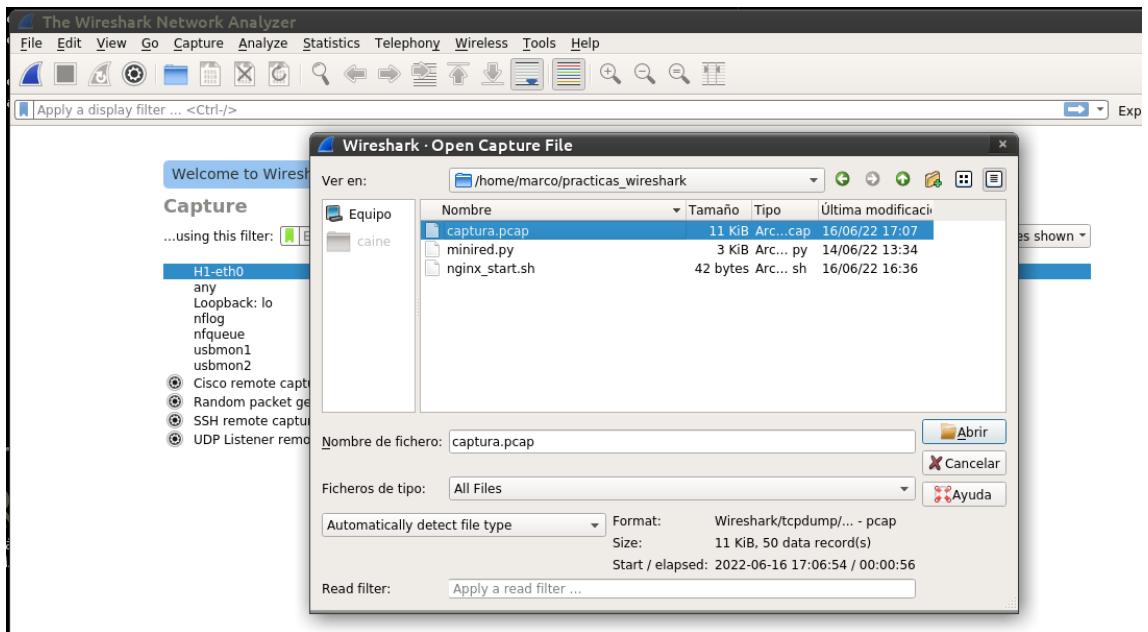
Parte 2:

En esta parte analizaremos los paquetes capturados con tcpdump en Wireshark.

- g) En el nodo: H1, ingrese **wireshark &** para iniciar Wireshark (la advertencia emergente no es importante para este laboratorio). Haga clic en Aceptar para continuar.

```
"Node: H1" (como superusuario)
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ 
marco@marco-VirtualBox:~/practicas_wireshark$ wireshark &
```

- h) En Wireshark, haga clic en File > Open (Archivo > Abrir). Seleccionen el archivo pcap guardado en la Parte 1.



- i) Aplique un filtro **tcp** a la captura. En este ejemplo las 3 primeras tramas son el tráfico que nos interesa.

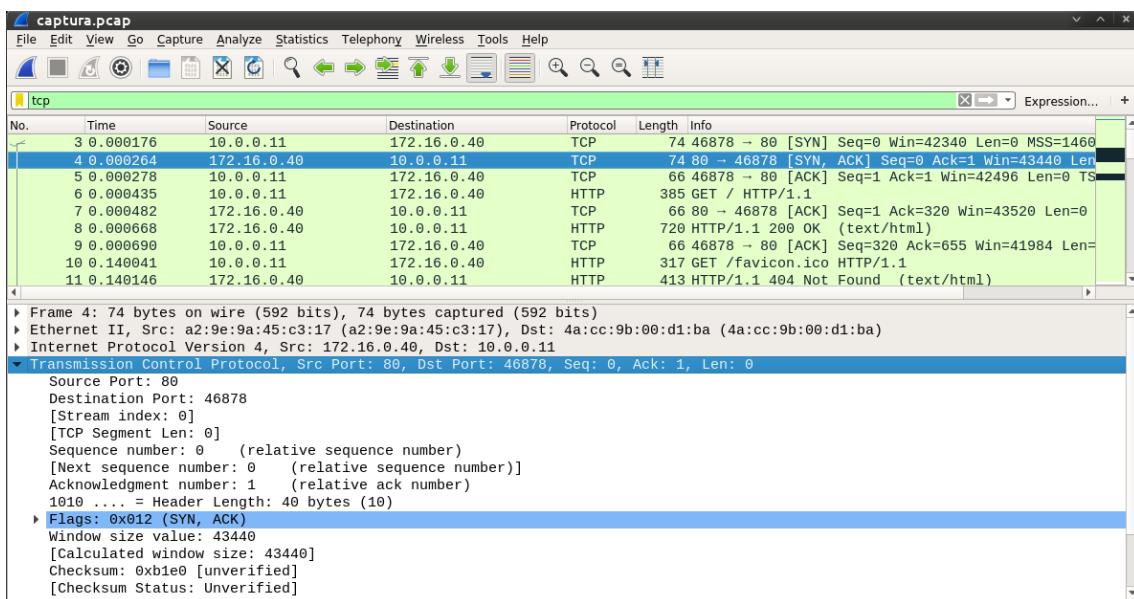
No.	Time	Source	Destination	Protocol	Length	Info
3	0.000176	10.0.0.11	172.16.0.40	TCP	74	46878 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460
4	0.000264	172.16.0.40	10.0.0.11	TCP	74	80 → 46878 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0
5	0.000278	10.0.0.11	172.16.0.40	TCP	66	46878 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TS

- j) En este ejemplo, la trama 1 es el inicio del protocolo de enlace de tres pasos entre el equipo H1 y el servidor en H4. En el panel de la lista de paquetes (sección superior de la ventana principal), seleccione el primer paquete, si es necesario.
k) Haga clic en la flecha que se encuentra a la izquierda del protocolo de control de transmisión en el panel de detalles del paquete para ampliar la vista y examinar la información de TCP. Localice la información de los puertos de origen y destino.

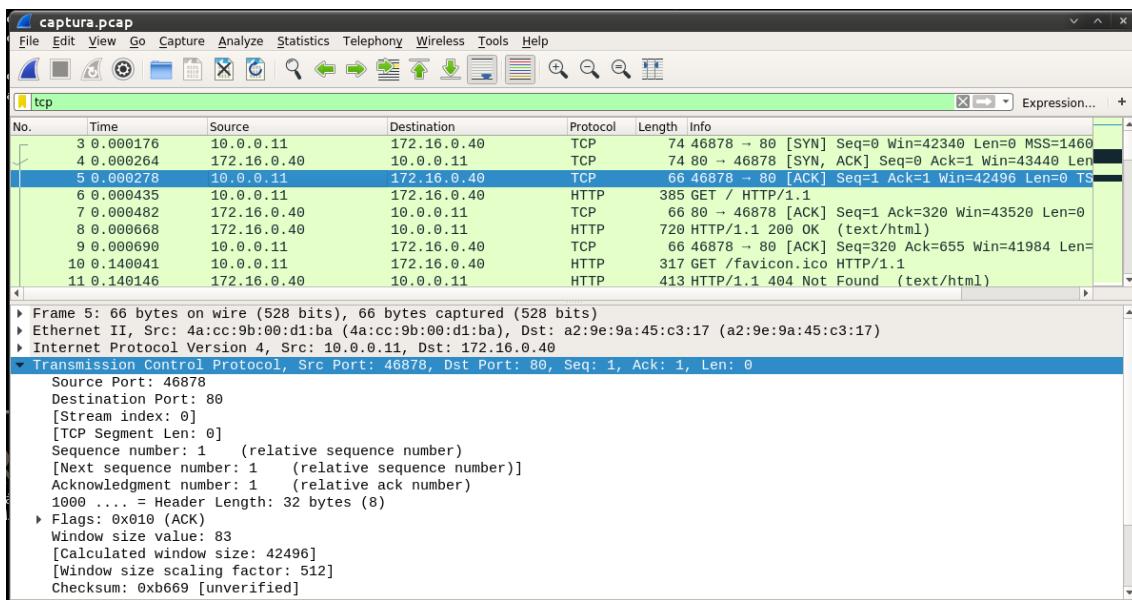
No.	Time	Source	Destination	Protocol	Length	Info
3	0.000176	10.0.0.11	172.16.0.40	TCP	74	46878 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460
4	0.000264	172.16.0.40	10.0.0.11	TCP	74	80 → 46878 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0
5	0.000278	10.0.0.11	172.16.0.40	TCP	66	46878 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TS
6	0.000435	10.0.0.11	172.16.0.40	HTTP	385	GET / HTTP/1.1
7	0.000482	172.16.0.40	10.0.0.11	TCP	66	80 → 46878 [ACK] Seq=1 Ack=320 Win=43520 Len=0
8	0.000668	172.16.0.40	10.0.0.11	HTTP	720	HTTP/1.1 200 OK (text/html)
9	0.000690	10.0.0.11	172.16.0.40	TCP	66	46878 → 80 [ACK] Seq=320 Ack=655 Win=41984 Len=0
10	0.140041	10.0.0.11	172.16.0.40	HTTP	317	GET /favicon.ico HTTP/1.1
11	0.140146	172.16.0.40	10.0.0.11	HTTP	413	HTTP/1.1 404 Not Found (text/html)

Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: 4a:cc:9b:00:d1:ba (4a:cc:9b:00:d1:ba), Dst: a2:9e:9a:45:c3:17 (a2:9e:9a:45:c3:17)
Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
Transmission Control Protocol, Src Port: 46878, Dst Port: 80, Seq: 0, Len: 0
Source Port: 46878
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0

- l) Haga clic en la flecha que se encuentra a la izquierda de los Flags. Un valor de 1 significa que el flag está definido. Localice el flag que está definido en este paquete.
- m) ¿Cuál es el número de puerto de origen de TCP?
- n) ¿Cómo clasificaría el puerto de origen?
- o) ¿Cuál es el número de puerto de destino de TCP?
- p) ¿Cómo clasificaría el puerto de destino?
- q) ¿Qué flags están establecidos?
- r) ¿Qué número de secuencia relativa está establecido?
- s) Seleccione el siguiente paquete en el protocolo de enlace de tres pasos. En este ejemplo, es la trama 2. Este es el servidor web que responde la solicitud inicial para iniciar una sesión.



- t) ¿Cuáles son los valores de los puertos origen y destino de TCP?
- u) ¿Qué flags están establecidos?
- v) ¿En qué valores están definidos los números de secuencia relativa y confirmación?
- w) Finalmente, seleccione el tercer paquete en el protocolo de enlace de tres pasos.



- x) ¿Qué flags están establecidos en el tercer paquete?
- y) ¿En qué valores están definidos los números de secuencia relativa y confirmación en el tercer paquete (si ambos tuviesen valor 1 la comunicación se ha establecido)?

Parte 3:

Investiga cómo puede analizar el archivo de captura con la propia herramienta tcpdump. Busque ayuda en el man sobre qué opción/es te permite indicar al comando de qué fichero debe tomar la información a procesar, el protocolo en el cual desea centrarse y el número máximos de paquetes a visualizar.

- z) ¿Cuáles son dichas opciones y qué propósito tienen?
- aa) En la ventana desde que lanzó el script de **minired.py**, escriba **quit** para detener Mininet.
- bb) Para limpiar todos los procesos que utilizó Mininet, escribe el comando **sudo mn -c** en el prompt.

3. Cuando utilizas Internet, estás utilizando el Sistema de Nombres de Dominio (DNS). DNS es una red distribuida de servidores que traduce nombres de dominio descriptivos como www.google.com a una dirección IP. Cuando se escribe la URL de un sitio web en el navegador, la PC realiza una consulta de DNS a la dirección IP del servidor DNS. La consulta del servidor DNS de su PC y la respuesta del servidor DNS hacen uso del Protocolo de Datagramas de Usuario (UDP) como protocolo de capa de transporte. A diferencia de TCP, UDP funciona sin conexión y no requiere una configuración de sesión. Las consultas y respuestas de DNS son muy pequeñas y no requieren la sobrecarga de TCP.

Objetivos de la práctica:

- En esta práctica de laboratorio, establecerá comunicación con un servidor DNS enviando una consulta de DNS mediante el protocolo de transporte UDP. Utilizará Wireshark para examinar los intercambios de consulta y respuesta de DNS con el mismo servidor.

Parte 1: Registrar la información sobre la configuración IP de la máquina virtual.

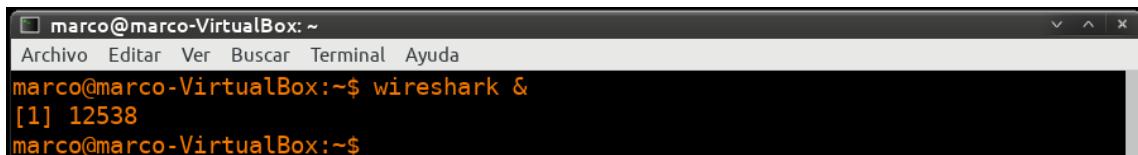
- a) En su máquina virtual de CAINE utilice los comandos necesarios para encontrar y registrar las direcciones IPv4 y MAC de las tarjetas de interfaz de red (NIC) virtuales de sus VM, la dirección IPv4 del gateway. Registre esta información en la tabla proporcionada.

Descripción	Configuración
Dirección IPv4	
Dirección MAC	
Dirección IPv4 de la pasarela	
Dirección IPv4 del servidor DNS	

Para ello explore la información que le pueden proporcionar los comandos **lshw**, **ip**, **netstat** y **nmcli**.

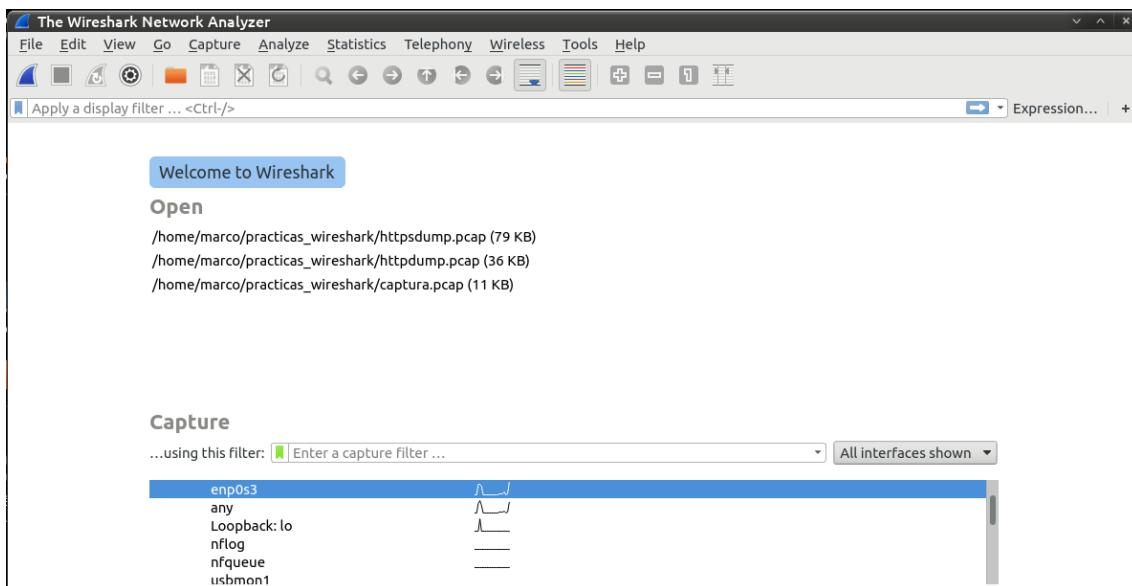
Parte 2:

- b) En una ventana de terminal introduce **wireshark &** para iniciar Wireshark. Haga clic en Aceptar para continuar.



```
marco@marco-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
marco@marco-VirtualBox:~$ wireshark &
[1] 12538
marco@marco-VirtualBox:~$
```

- c) En la ventana de Wireshark selecciona con doble clic, en el apartado Captura, el interfaz desde el cual va a capturar los paquetes.



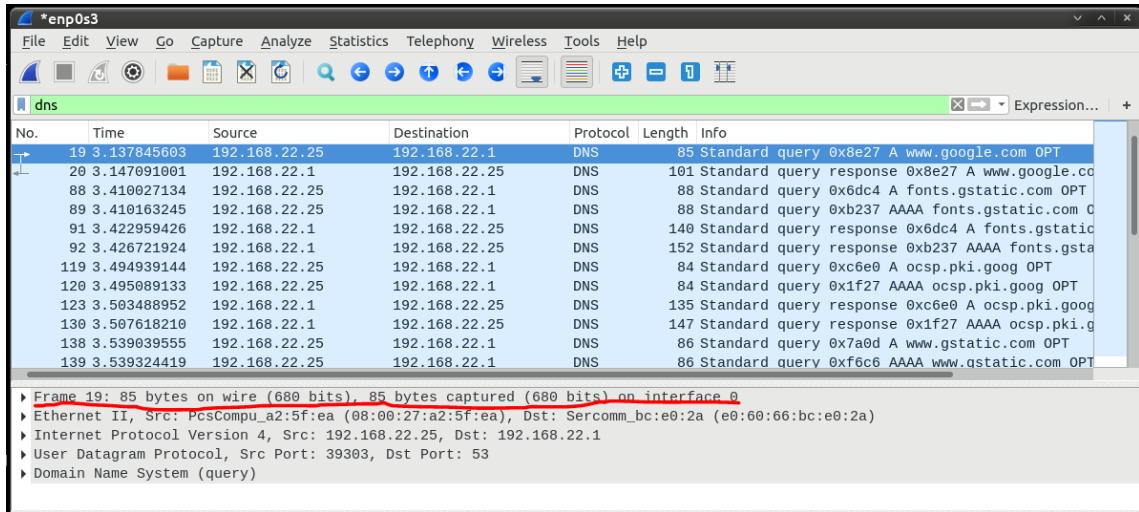
- d) Abre el navegador web y dirígete a **www.google.com**.
- e) Haga clic en **Stop** (Detener) para detener la captura de Wireshark cuando vea la página de inicio de Google.

Parte 3: Analizar los paquetes capturados de DNS y UDP.

- f) En la ventana principal de Wireshark, escriba **dns** en el campo Filter (Filtro). Haga clic en **Apply** (Aplicar).

No.	Time	Source	Destination	Protocol	Length	Info
19	3.137845603	192.168.22.25	192.168.22.1	DNS	85	Standard query 0x8e27 A www.google.com OPT
20	3.147091001	192.168.22.1	192.168.22.25	DNS	101	Standard query response 0x8e27 A www.google.co
88	3.410027134	192.168.22.25	192.168.22.1	DNS	88	Standard query 0x6dc4 A fonts.gstatic.com OPT
89	3.410163245	192.168.22.25	192.168.22.1	DNS	88	Standard query 0xb237 AAAA fonts.gstatic.com C
91	3.422959426	192.168.22.1	192.168.22.25	DNS	140	Standard query response 0x6dc4 A fonts.gstatic
92	3.426721924	192.168.22.1	192.168.22.25	DNS	152	Standard query response 0xb237 AAAA fonts.gstatic
119	3.494939144	192.168.22.25	192.168.22.1	DNS	84	Standard query 0xc6e0 A ocsp.pki.goog OPT
120	3.495089133	192.168.22.25	192.168.22.1	DNS	84	Standard query 0x1f27 AAAA ocsp.pki.goog OPT
123	3.503488952	192.168.22.1	192.168.22.25	DNS	135	Standard query response 0xc6e0 A ocsp.pki.goog
130	3.507618210	192.168.22.1	192.168.22.25	DNS	147	Standard query response 0x1f27 AAAA ocsp.pki.g
138	3.539039555	192.168.22.25	192.168.22.1	DNS	86	Standard query 0x7a0d A www.gstatic.com OPT
139	3.539324419	192.168.22.25	192.168.22.1	DNS	86	Standard query 0xf6c6 AAAA www.gstatic.com OPT

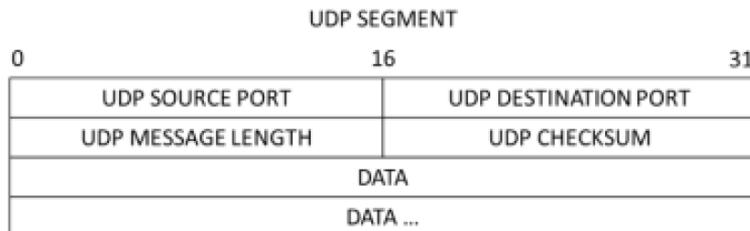
- g) En el panel de lista de paquetes (sección superior) de la ventana principal, localice el paquete que incluye **Standard query** (Consulta estándar) y **A www.google.com**. Observe la trama 19 anterior como ejemplo.
- h) Los campos del paquete, resaltados en color gris, se muestran en el panel de detalles del paquete (sección media) de la ventana principal. En la primera línea del panel de detalles del paquete, la trama 19 tiene 85 bytes de datos transmitidos (on wire). Esta es la cantidad de bytes que se necesitó para enviar una consulta DNS a un servidor con nombre que está solicitando las direcciones IP de www.google.com. Si utilizaste otra dirección web, la cantidad de bytes podría ser diferente.



- i) La línea Ethernet II muestra las direcciones MAC de origen y destino. La dirección MAC de origen proviene de su máquina virtual porque su máquina virtual fue la que originó la consulta DNS. La dirección MAC de destino proviene del gateway predeterminado porque esta es la última parada antes de que esta consulta salga de la red local. ¿Es la dirección MAC de origen la misma que la registrada en la Parte 1 para la VM?
- j) En la línea del Protocolo de Internet Versión 4 (IPv4), la captura del paquete IP Wireshark indica que la dirección IP de origen de esta consulta de DNS es 192.168.22.25 (en este ejemplo) y la dirección IP de destino es 192.168.22.1 (en este ejemplo). ¿Puede identificar la dirección IP y dirección MAC de origen y destino de este paquete?

Dispositivo	Dirección IP	Dirección MAC
Máquina virtual cliente		
Destino servidor DNS/Gateway predeterminado		

- k) El paquete IP y el encabezado encapsulan el segmento de UDP. El segmento de UDP contiene la consulta de DNS como datos. Haga clic en la flecha contigua a User Datagram Protocol para ver los detalles. Observa que solo hay cuatro campos. El número del puerto de origen en este ejemplo es 39303. La MV generó de manera aleatoria el puerto de origen utilizando números de puerto que no están reservados. El puerto de destino es 53. El puerto 53 es un puerto conocido reservado para el uso con DNS. Los servidores DNS esperan en el puerto 53 las consultas de DNS de los clientes.



```

▶ Frame 19: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_a2:5f:ea (08:00:27:a2:5f:ea), Dst: Sercomm_bc:e0:2a (e0:60:66:bc:e0:2a)
▶ Internet Protocol Version 4, Src: 192.168.22.25, Dst: 192.168.22.1
▶ User Datagram Protocol, Src Port: 39303, Dst Port: 53
  Source Port: 39303
  Destination Port: 53
  Length: 51
  Checksum: 0xadaf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
▶ Domain Name System (query)

```

En este ejemplo, la longitud del segmento de UDP es de 51 bytes. La longitud del segmento UDP de su ejemplo puede ser diferente. De los 51 bytes, 8 bytes se utilizan como encabezado. Los datos de la consulta de DNS utilizan los otros 43 bytes. Los 43 bytes de los datos de consulta DNS están el panel de bytes del paquete (sección inferior) de la ventana principal de Wireshark.

```

▶ User Datagram Protocol, Src Port: 39303, Dst Port: 53
  ▶ Domain Name System (query)
    Transaction ID: 0x8e27
    ▶ Flags: 0x0100 Standard query
      0... .... .... = Response: Message is a query
      .000 0... .... = Opcode: Standard query (0)
      .... 0. .... = Truncated: Message is not truncated
      .... .1 .... = Recursion desired: Do query recursively
      .... .... 0.... = Z: reserved (0)
      .... .... .0.... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
    ▶ Queries
    ▶ Additional records
      [Response In: 20]

```

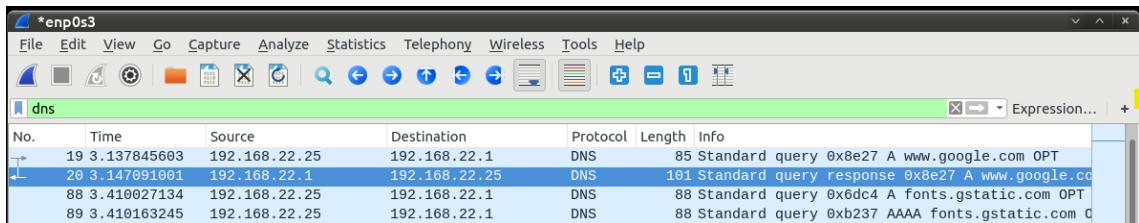
En este ejemplo, la dirección de destino es la del servidor DNS.

- Haz clic en la flecha que se encuentra a la izquierda de los Flags. Un valor de 1 significa que el flag está definido. Localice el flag que está definido en este paquete.
- El checksum es usado para determinar la integridad del encabezado de UDP después de haber atravesado Internet. El encabezado de UDP tiene poca sobrecarga porque UDP no tiene campos que estén asociados con el protocolo de enlace de tres vías en TCP. Cualquier problema de confiabilidad de la transferencia de datos que ocurra debe ser manejado por la capa de aplicación. Expanda lo necesario para ver los detalles. Registre sus resultados de Wireshark en la tabla siguiente:

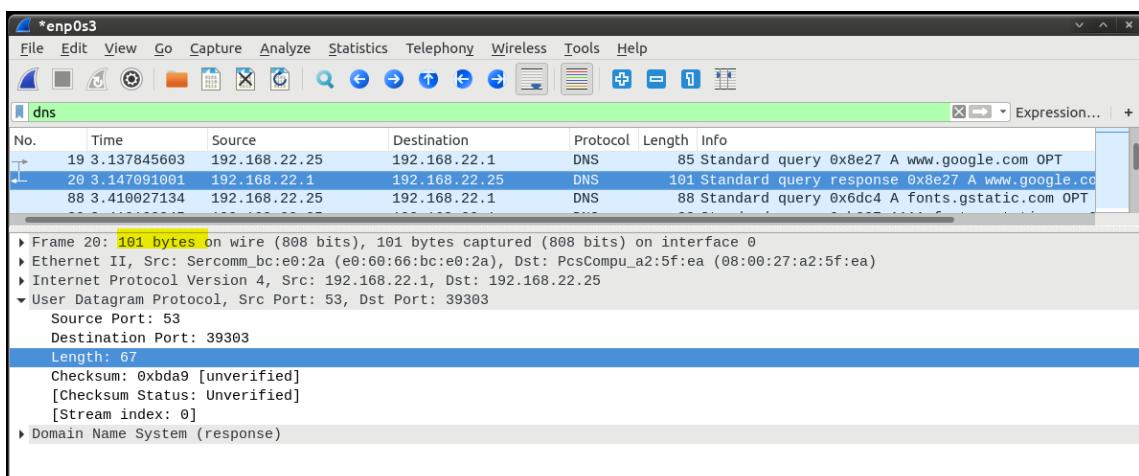
Descripción	Resultado Wireshark
Tamaño de la trama	
MAC origen	
MAC destino	
IP origen	
IP destino	
Puerto origen	
Puerto destino	

- n) ¿Es la dirección IP de origen la misma que la dirección IP de la MV que registró en la parte 1?
- o) ¿Es la dirección IP de destino la misma que la puerta de enlace predeterminada (gateway) que observó en la parte 1?

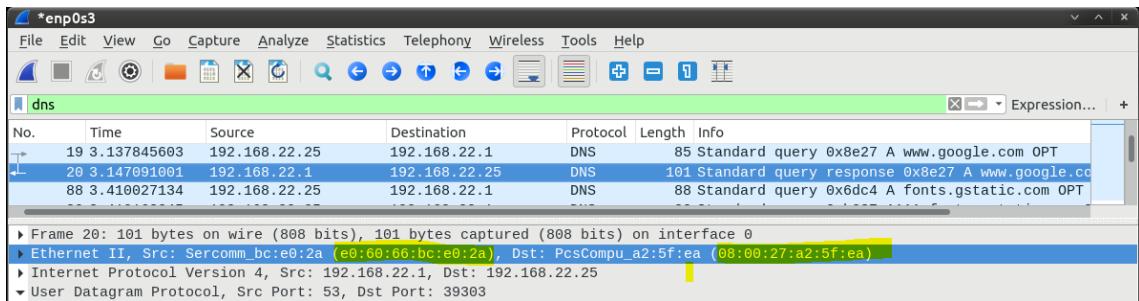
Parte 4: Examinar los campos de un paquete de respuesta DNS.



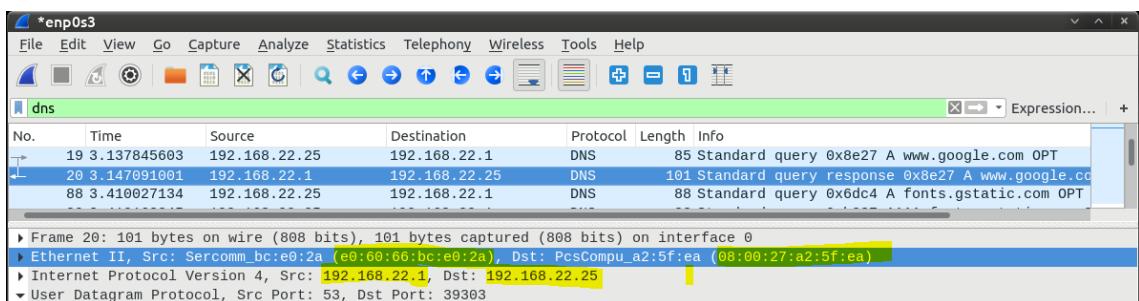
En este ejemplo, la trama 20 es el paquete de respuesta DNS correspondiente. Observe que la cantidad de bytes en la conexión es 101. Es un paquete más grande en comparación con el paquete de consulta de DNS. Esto se debe a que el paquete de respuesta DNS incluirá información variada sobre el dominio.



- p) En la trama Ethernet II para la respuesta de DNS, ¿qué dispositivo es la dirección MAC de origen y qué dispositivo es la dirección MAC de destino?



- q) Observe las direcciones IP de origen y destino en este paquete IP. ¿Cuál es la dirección IP de destino? ¿Cuál es la dirección IP de origen? ¿Qué sucedió con los roles de origen y destino correspondientes a la VM y al gateway predeterminado?



- r) En el segmento UDP, el rol de los números de puerto también se invirtió. El número del puerto de destino es 39303. El número de puerto 39303 es el mismo puerto que generó la MV cuando se envió la consulta DNS al servidor DNS. La MV espera una respuesta DNS en este puerto. El número del puerto de origen es 53. El servidor DNS espera una consulta de DNS en el puerto 53 y luego envía una respuesta de DNS con un número de puerto de origen 53 al originador de la consulta de DNS. Al expandirse la respuesta de DNS, observa las direcciones IP resueltas para www.google.com en la sección Answers (Respuestas) y captura la pantalla resaltando dicha información.
4. El Protocolo de transferencia de hipertexto (HyperText Transfer Protocol, HTTP) es un protocolo de la capa de aplicación que presenta datos a través de un navegador web. Con HTTP, no se protegen los datos intercambiados entre dos dispositivos que se están comunicando.

Con HTTPS, se emplea cifrado por medio de un algoritmo matemático. Este algoritmo oculta el verdadero significado de los datos que se está intercambiando. Esto último se hace mediante el uso de certificados.

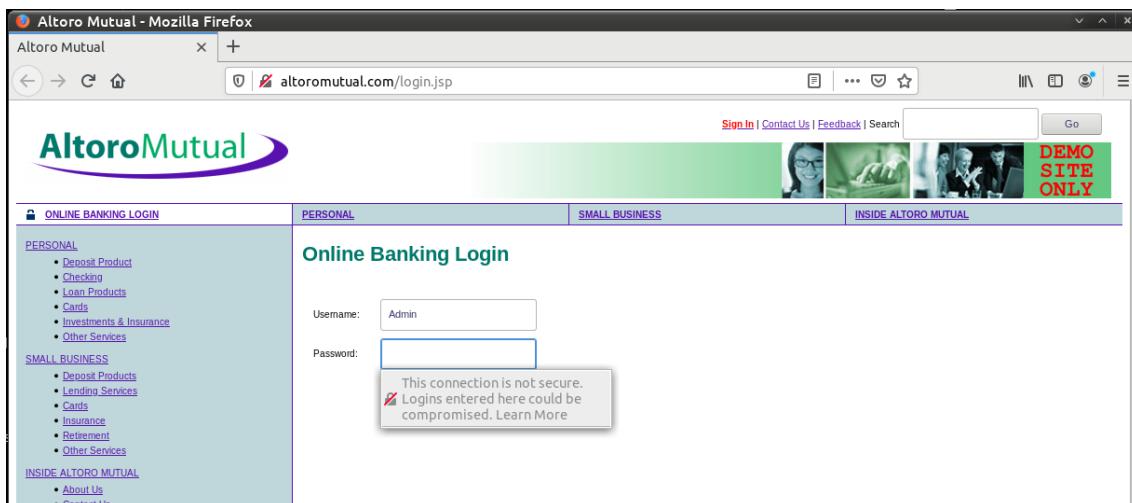
Independientemente de que se utilice HTTP o HTTPS, solo se recomienda intercambiar datos con sitios web de confianza. **El solo hecho de que un sitio utilice HTTPS no significa que sea confiable. Los atacantes suelen utilizar HTTPS para ocultar sus actividades.**

En este ejercicio se capturará tráfico HTTP y HTTPS y se analizará con Wireshark.

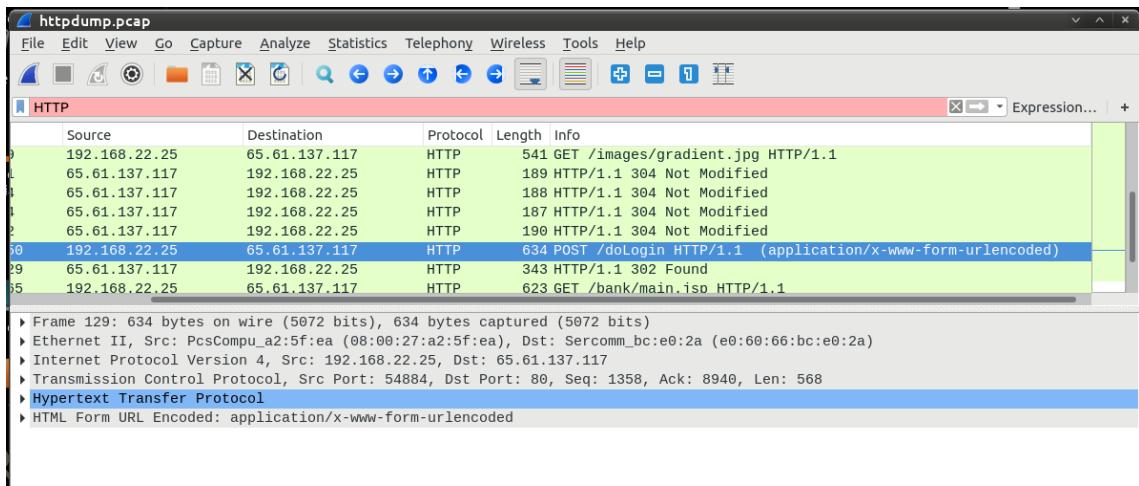
Parte 1: Capturar y ver tráfico HTTP

En la máquina virtual en la que ha instalado CAINE utilizarás el comando **tcpdump** (también podrías hacerlo desde Wireshark) para capturar el contenido del tráfico HTTP. Utilizarás las opciones del comando **tcpdump** para guardar el resultado de la captura en un archivo **.pcap** que luego se analizará más tarde desde Wireshark.

- En una terminal averigua la configuración de las interfaces de red.
- Busque información del comando **tcpdump**. Concretamente busque información sobre las opciones del comando que le permiten elegir el nombre lógico del interfaz de red cuyo tráfico va a ser capturado, así como la opción que permite indicar en qué fichero **.pcap** se registrará dicho tráfico.
- Desde la misma terminal introduzca el comando **tcpdump** y registre todo el tráfico que genere la interfaz desde ese momento. Almacene dicho tráfico en un fichero denominado **httpdump.pcap**. Ejecute el comando con privilegios de administrador.
- Abra Firefox. En la barra de navegación escriba la dirección <http://www.altoromutual.com/login.jsp>. Este sitio web utiliza tráfico no cifrado. En el campo **Username** escribe **Admin** y lo mismo en el campo **Password**. Luego pulsa el botón login para iniciar sesión.



- Una vez lo hayas hecho, cierra el navegador web. Finaliza la captura de **tcpdump** pulsando **CTRL-C** desde la terminal donde lo lanzaste.
- Abre la aplicación Wireshark. Carga el archivo con la captura anterior. Pon en filtro el protocolo HTTP y busca la primera trama correspondiente al verbo HTTP **POST**.

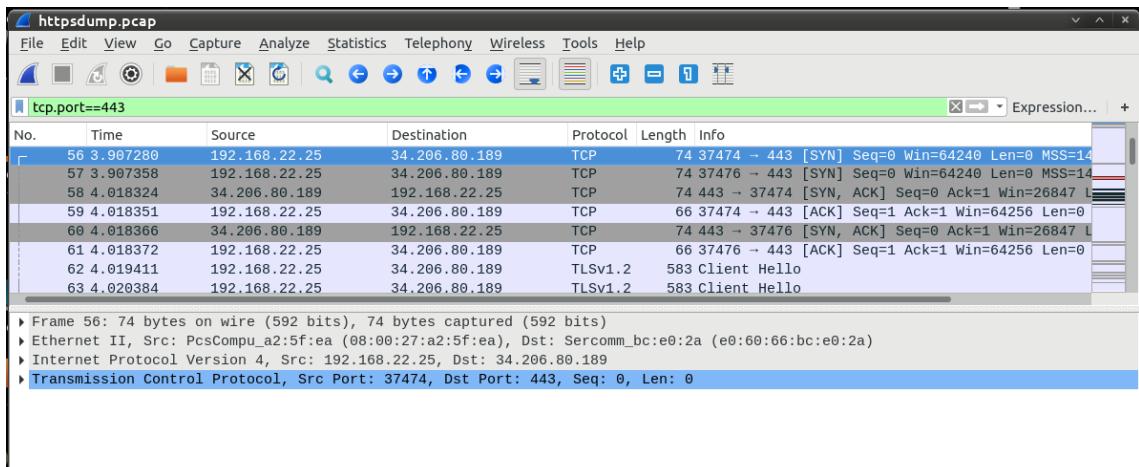


- g) El mensaje aparece en la parte media de la pantalla. Expanda la sección HTML Form. ¿Qué datos aparecen?
- h) Cierra la aplicación Wireshark.

Parte 2: Capturar y ver tráfico HTTPS

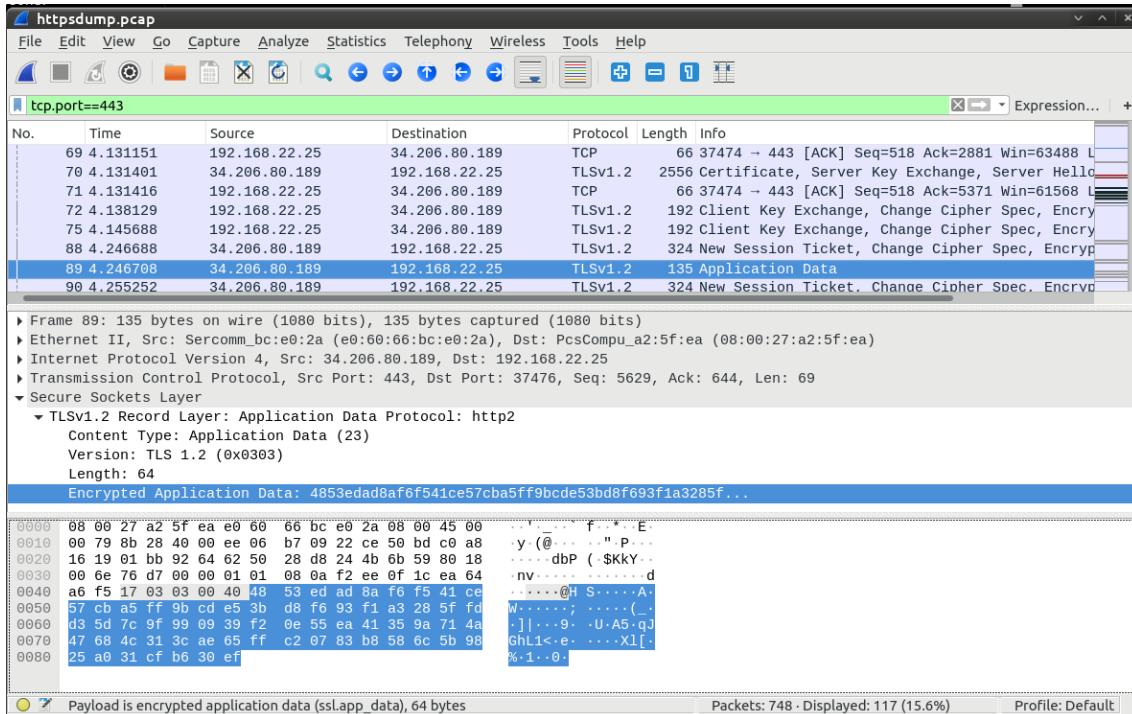
Vamos a utilizar el comando `tcpdump` para capturar el contenido del tráfico HTTPS siguiendo el mismo procedimiento que seguimos para capturar el tráfico HTTP. El fichero donde almacenarás el tráfico capturado se denominará **httpsdump.pcap**. La página a la que te conectarás desde el navegador será la página <https://httpbin.org/forms/post>. Rellena el formulario que te presenta dicha página con la información que quieras. Tan pronto envíes el formulario, finaliza la captura.

- i) Abre la aplicación Wireshark. Carga el archivo con la captura anterior. Filtra el tráfico HTTPS por el puerto 443, para ello pon en el filtro `tcp.port==443` y aplícalo.



- j) Desplázate por los diferentes mensajes HTTPS y selecciona el mensaje **Application Data**.

- k) ¿Qué ha reemplazado a la sección HTTP que estaba en el archivo de captura anterior?
- l) Expande completamente la sección **Secure Socket Layer**. Haz clic en Encrypted Application Data. Los datos que introdujiste en el formulario, ¿son legibles o están encriptados?



5. Dos de los protocolos de la capa de transporte de TCP/IP son TCP (definido en RFC 761) y UDP (definido en RFC 768). Los dos protocolos admiten la comunicación de protocolos de capa superior. Por ejemplo, TCP se utiliza para proporcionar soporte de capa de transporte para el protocolo de transferencia de hipertexto (HTTP) y transferencia de ficheros (FTP), entre otros. UDP proporciona soporte de capa de transporte para el sistema de nombres de dominio (DNS) y TFTP (Trivial File Transfer Protocol), entre otros.

Objetivos de la práctica:

- En esta ejercicio se utilizará Wireshark para capturar y analizar campos de encabezado del protocolo TCP para las transferencias de archivos FTP entre el equipo host y un servidor FTP anónimo. Se utilizará la línea de comandos del terminal para establecer una conexión a un servidor FTP anónimo y descargar un archivo.

Paso 1: Iniciar una captura de Wireshark.

- a) En su máquina virtual de CAINE utilice los comandos necesarios para averiguar y registrar las direcciones IPv4 y MAC de las tarjetas de interfaz de red (NIC) virtuales de sus VM, la dirección IPv4 del gateway. Registre esta información en la tabla proporcionada.

Descripción	Configuración
Nombre lógico	
Dirección IPv4	
Dirección MAC	
Dirección IPv4 de la pasarela	
Dirección IPv4 del servidor DNS	

Para ello explore la información que le pueden proporcionar los comandos **lshw**, **ip**, **netstat** y **nmcli**.

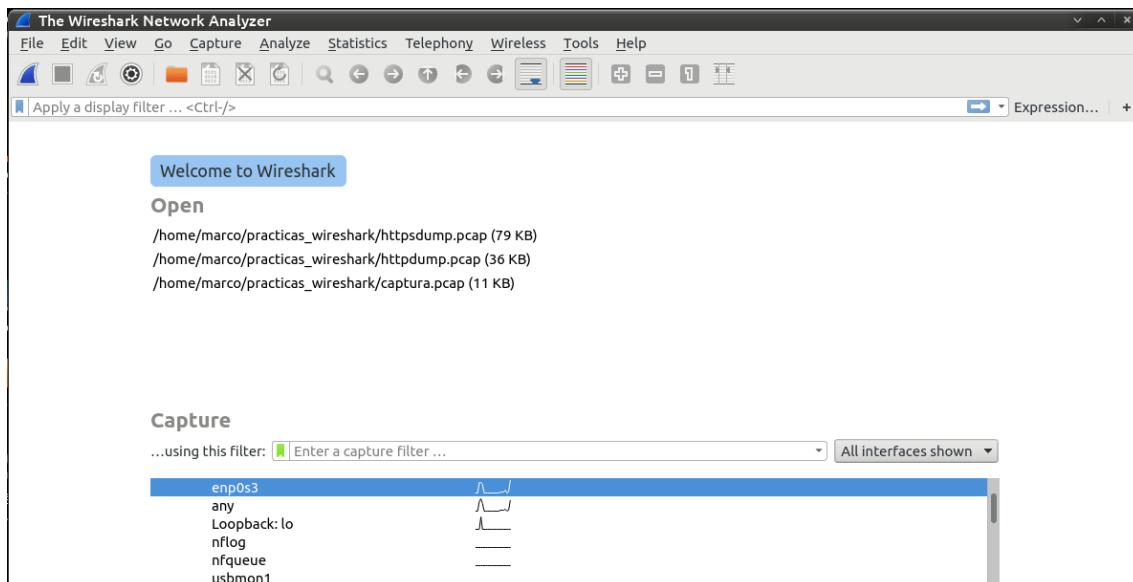
Paso 2:

- b) En una ventana de terminal introduce **wireshark &** para iniciar Wireshark. Haga clic en Aceptar para continuar.

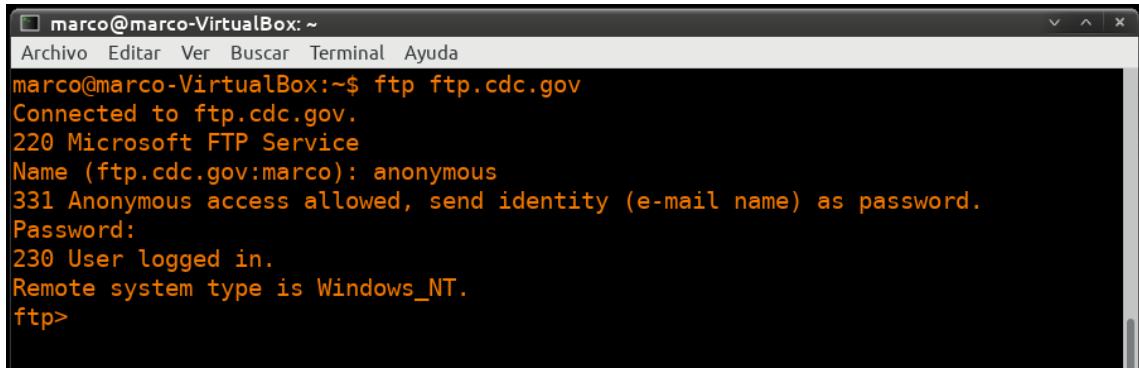


```
marco@marco-VirtualBox:~$ wireshark &
[1] 12538
marco@marco-VirtualBox:~$
```

- c) En la ventana de Wireshark selecciona con doble clic, en el apartado Captura, el interfaz desde el cual va a capturar los paquetes.



- d) Abra otra ventana del terminal para acceder al sitio ftp externo. Escriba **ftp ftp.cdc.gov** en el cursor. Conéctense al sitio FTP de los Centros para el Control y la Prevención de Enfermedades (CDC) con el usuario **anonymous** y sin contraseña.

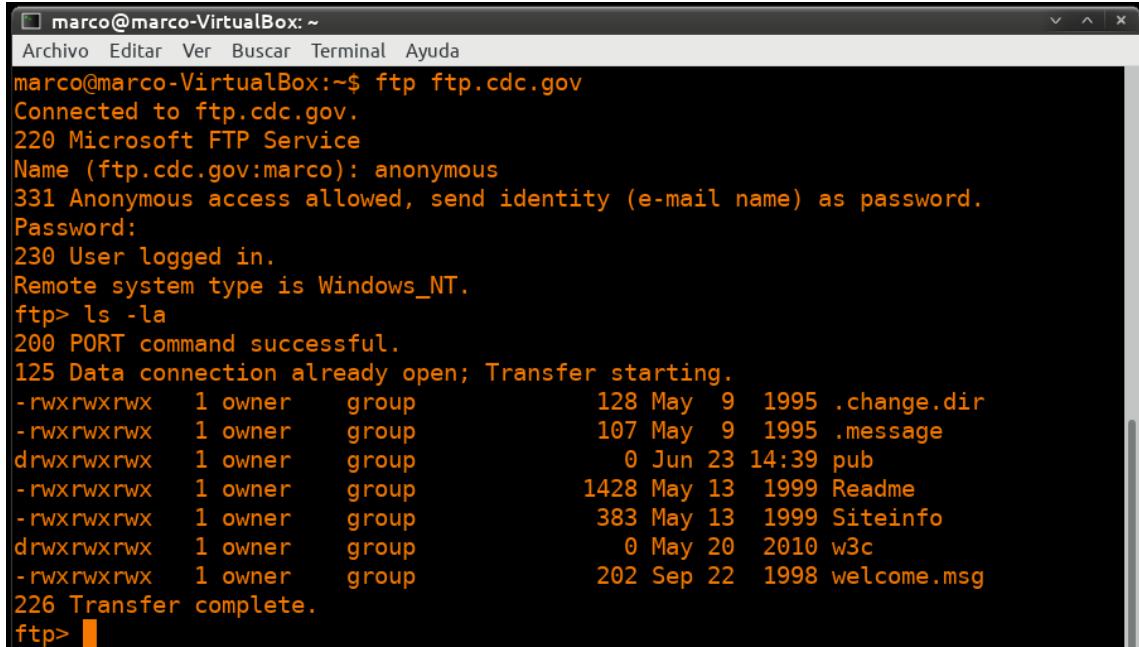


```
marco@marco-VirtualBox:~$ ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
Name (ftp.cdc.gov:marco): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp>
```

- e) Haga clic en **Stop** (Detener) para detener la captura de Wireshark cuando vea la página de inicio de Google.

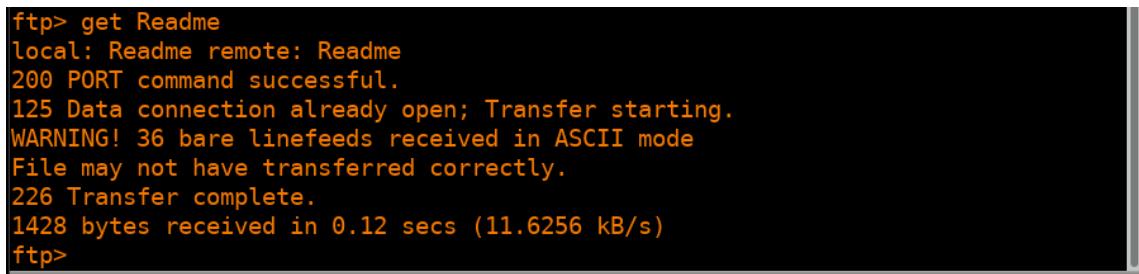
Paso 3: Descargar el archivo **Readme**.

- f) Localiza y descarga el archivo Readme; para ello, introduce el comando **ls -la** para generar una lista de los archivos.



```
marco@marco-VirtualBox:~$ ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
Name (ftp.cdc.gov:marco): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls -la
200 PORT command successful.
125 Data connection already open; Transfer starting.
-rwxrwxrwx 1 owner group 128 May  9 1995 .change.dir
-rwxrwxrwx 1 owner group 107 May  9 1995 .message
drwxrwxrwx 1 owner group 0 Jun 23 14:39 pub
-rwxrwxrwx 1 owner group 1428 May 13 1999 Readme
-rwxrwxrwx 1 owner group 383 May 13 1999 Siteinfo
drwxrwxrwx 1 owner group 0 May 20 2010 w3c
-rwxrwxrwx 1 owner group 202 Sep 22 1998 welcome.msg
226 Transfer complete.
ftp> ■
```

- g) Introduzca el comando **get Readme** para descargar el archivo. Una vez finalizada la transferencia, introduzca **quit** para salir de ftp.



```
ftp> get Readme
local: Readme remote: Readme
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 36 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
1428 bytes received in 0.12 secs (11.6256 kB/s)
ftp>
```

Paso 4: Detener la captura de Wireshark.

- h) Wireshark capturó muchos paquetes durante la sesión FTP para ftp.cdc.gov. Si quiere limitar la cantidad de datos para el análisis, apliquen el filtro **tcp and ip.addr == 198.246.117.106** y haga clic en **Apply** (Aplicar). **NOTA:** La dirección IP **198.246.117.106**, era la dirección correspondiente a **ftp.cdc.gov** cuando se creó esta práctica de laboratorio, puede haber cambiado en el momento de realizarla. Sus direcciones IP pueden ser diferentes. Si hubiese cambiado, busca en el primer paquete TCP que inició el Protocolo de enlace de 3 vías con ftp.cdc.gov. La dirección IP de destino es la dirección IP que se debe utilizar para el filtro.

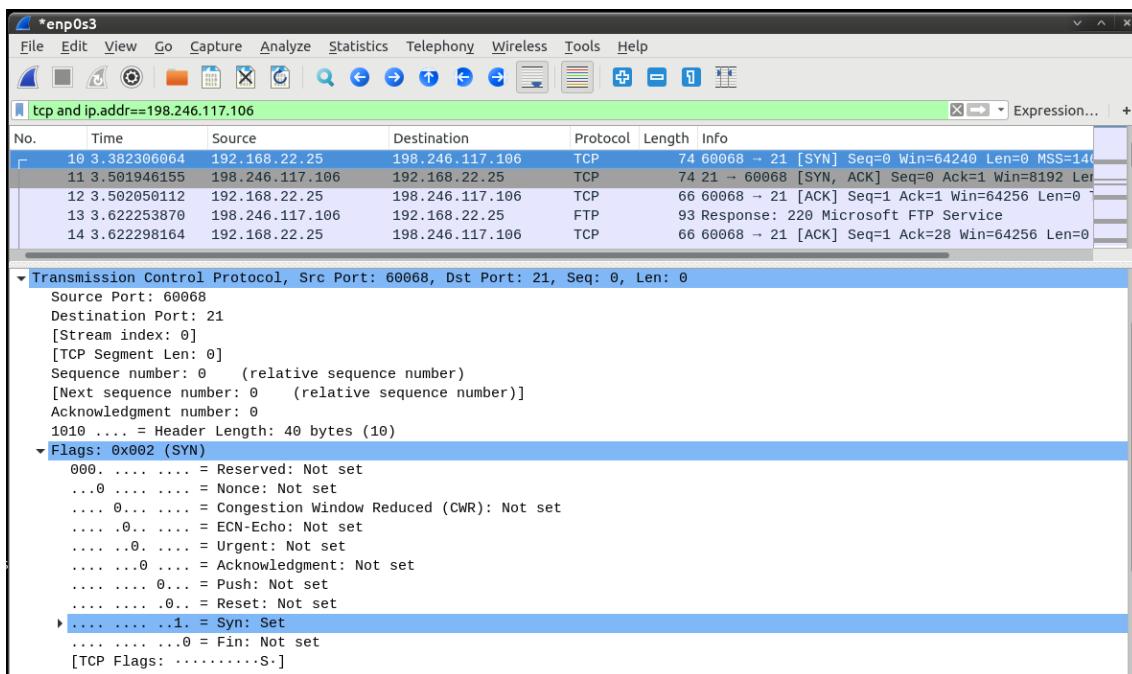
Paso 5: Analizar los paquetes TCP.

Después que el filtro TCP ha sido aplicado, los primeros tres paquetes (sección de arriba) muestran la secuencia de [SYN], [SYN, ACK], y [ACK] que es el protocolo de enlace de tres vías de TCP.

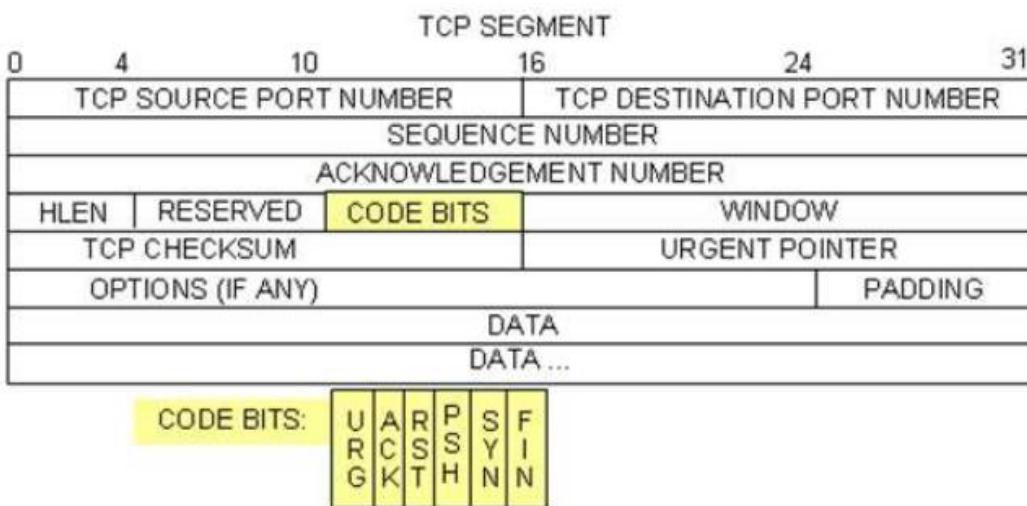
No.	Time	Source	Destination	Protocol	Length	Info
10	3.382306064	192.168.22.25	198.246.117.106	TCP	74	60068 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1464
11	3.501946155	198.246.117.106	192.168.22.25	TCP	74	21 → 60068 [SYN, ACK] Seq=1 Ack=1 Win=8192 Len=0
12	3.502050112	192.168.22.25	198.246.117.106	TCP	66	60068 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 T
13	3.622253870	198.246.117.106	192.168.22.25	FTP	93	Response: 220 Microsoft FTP Service
14	3.622298164	192.168.22.25	198.246.117.106	TCP	66	60068 → 21 [ACK] Seq=1 Ack=28 Win=64256 Len=0
15	8.963989087	192.168.22.25	198.246.117.106	FTP	82	Request: USER anonymous
16	9.084419961	198.246.117.106	192.168.22.25	FTP	138	Response: 331 Anonymous access allowed, send i
17	9.084489782	192.168.22.25	198.246.117.106	TCP	66	60068 → 21 [ACK] Seq=17 Ack=100 Win=64256 Len=0
18	14.919257743	192.168.22.25	198.246.117.106	FTP	88	Request: PASS marco@uniiovi.es
19	15.052547523	198.246.117.106	192.168.22.25	FTP	87	Response: 230 User logged in.
20	15.052636923	192.168.22.25	198.246.117.106	TCP	66	60068 → 21 [ACK] Seq=39 Ack=121 Win=64256 Len=0
21	15.053406691	192.168.22.25	198.246.117.106	FTP	72	Request: SYST
22	15.173516657	198.246.117.106	192.168.22.25	FTP	82	Response: 215 Windows_NT
23	15.217740261	192.168.22.25	198.246.117.106	TCP	66	60068 → 21 [ACK] Seq=45 Ack=137 Win=64256 Len=0
24	15.276944122	198.246.117.106	192.168.22.25	FTP	93	Request: PORT 192,168,22,25,220,83
25	17.277034209	192.168.22.25	198.246.117.106	TCP	66	20 → 56403 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
26	17.277034209	198.246.117.106	192.168.22.25	TCP	66	56403 → 20 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
27	17.277034209	192.168.22.25	198.246.117.106	TCP	66	56403 → 20 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0

TCP se utiliza en forma continua durante una sesión para controlar la entrega de datagramas, verificar la llegada de datagramas y administrar el tamaño de la ventana. Para cada intercambio de datos entre el cliente FTP y el servidor FTP, se inicia una nueva sesión TCP. Al término de la transferencia de datos, se cierra la sesión TCP. Cuando finaliza la sesión FTP, TCP realiza un cierre y un apagado ordenados.

En Wireshark, se encuentra disponible información detallada sobre TCP en el panel de detalles del paquete (sección media). Resalte el primer datagrama TCP del host, y expanda las porciones del datagrama de TCP, como se muestra a continuación.



El datagrama expandido de TCP parece similar al panel detallado del paquete, como se muestra a continuación.



La imagen anterior es un diagrama del datagrama TCP. Se proporciona una explicación de cada campo para referencia:

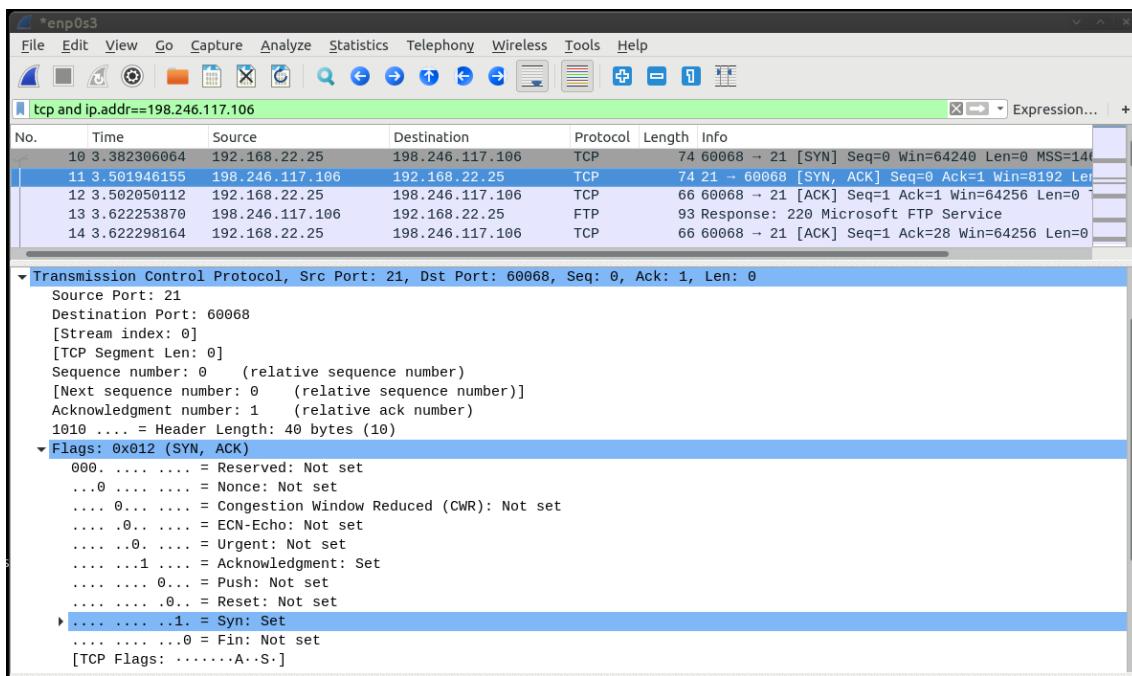
- El número de **puerto de origen TCP** pertenece al host de la sesión TCP que abrió una conexión. Generalmente el valor es un valor aleatorio superior a 1.023.
- El **número de puerto de destino TCP** se utiliza para identificar el protocolo de capa superior o la aplicación en el sitio remoto. Los valores en el intervalo de 0 a 1023 representan los “puertos bien conocidos” y están asociados a servicios y aplicaciones populares (como se describe en la RFC 1700), por ejemplo, Telnet, FTP y HTTP. La combinación de la dirección IP de origen, el puerto de origen, la

dirección IP de destino y el puerto de destino identifica de manera exclusiva la sesión para el remitente y para el destinatario.

- **Sequence number** (Número de secuencia) especifica el número del último byte en un segmento.
- **Acknowledgment number** (Número de reconocimiento) especifica el siguiente byte que espera el destinatario.
- **Code bits** (bits de código) tiene un significado especial en la administración de sesiones y en el tratamiento de los segmentos. Entre los valores interesantes se encuentran:
 - **ACK**: reconocimiento de la recepción de un segmento.
 - **SYN**: sincronizar, solo se define cuando se negocia una sesión de TCP nueva durante el protocolo de enlace de tres vías de TCP.
 - **FIN**: finalizar, la solicitud para cerrar la sesión de TCP.
- **Window size** (Tamaño de la ventana) es el valor de la ventana deslizante. Determina cuántos octetos pueden enviarse antes de esperar un reconocimiento.
- **Urgent pointer** (Puntero urgente) solo se utiliza con un marcador urgente (URG) cuando el remitente necesita enviar datos urgentes al destinatario.
- En **Options** (Opciones), hay una sola opción actualmente, y se define como el tamaño máximo del segmento TCP (valor opcional).
 - i) Utiliza la captura Wireshark del inicio de la primera sesión TCP (bit SYN fijado en 1) para completar la información acerca del encabezado TCP. Es posible que algunos campos no se apliquen a este paquete. De la MV al servidor CDC (solamente el bit SYN está definido en 1):

Descripción	Resultados de Wireshark
Dirección IP Origen	
Dirección IP Destino	
Puerto de Origen	
Puerto de Destino	
Número de secuencia	
Número de Reconocimiento	
Longitud del encabezado	
Tamaño de la ventana	

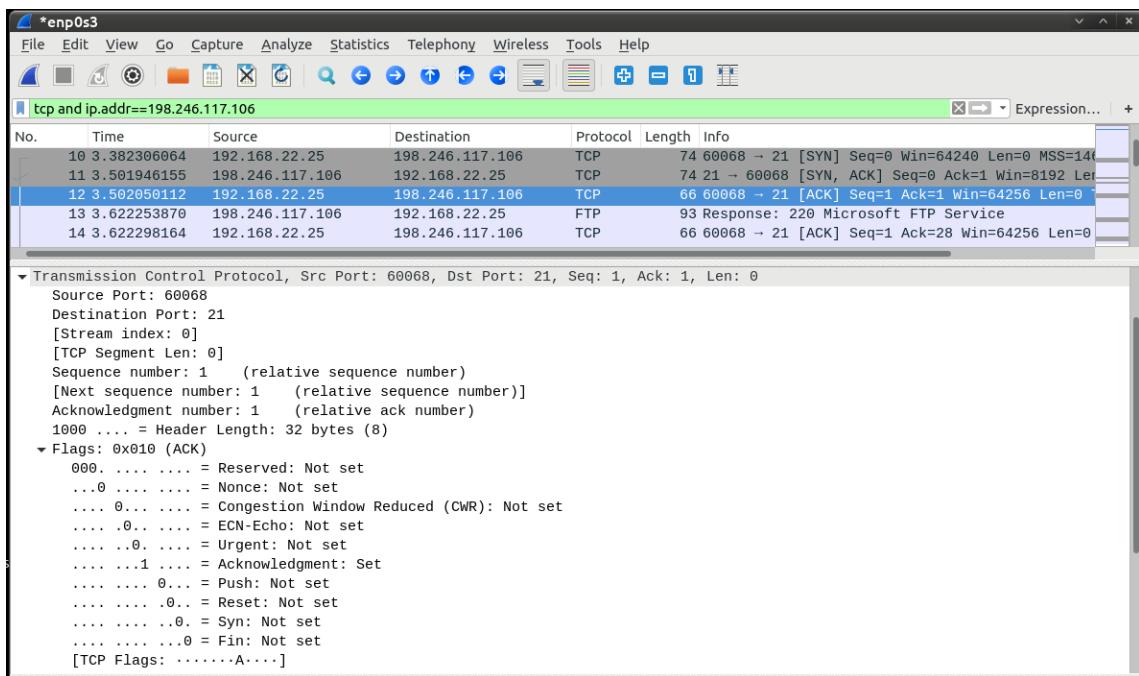
En la segunda captura filtrada de Wireshark, el servidor FTP del CDC confirma que recibió la solicitud de la MV. Observe los valores de los bits de SYN y ACK.



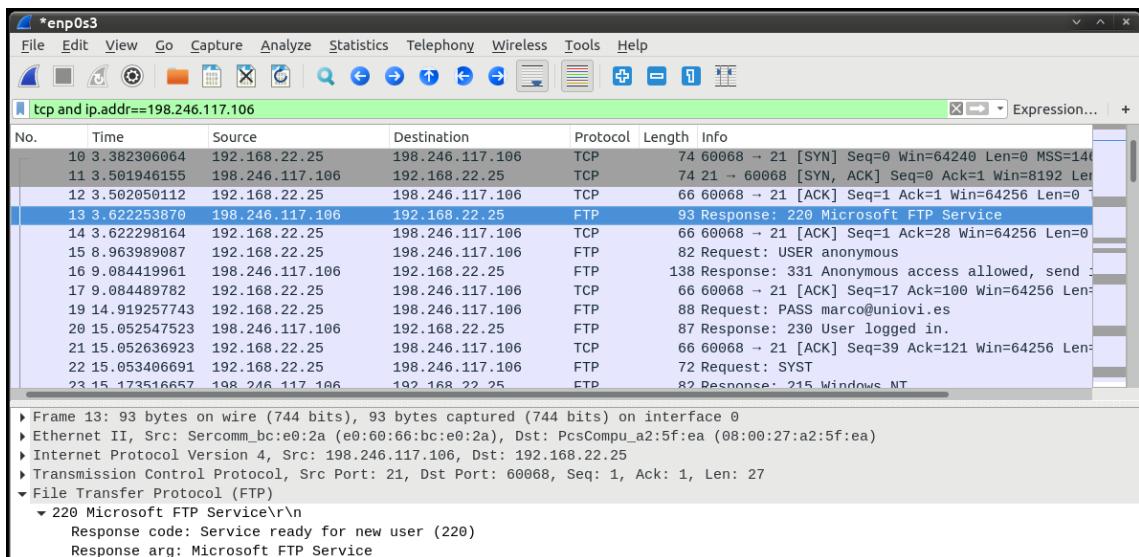
j) Complete la siguiente información sobre el mensaje de SYN-ACK.

Descripción	Resultados de Wireshark
Dirección IP Origen	
Dirección IP Destino	
Puerto de Origen	
Puerto de Destino	
Número de secuencia	
Número de Reconocimiento	
Longitud del encabezado	
Tamaño de la ventana	

En la etapa final de la negociación para establecer las comunicaciones, la VM envía un mensaje de acuse de recibo al servidor. Observen que solo el bit ACK está definido en 1, y que el número de secuencia se incrementó a 1.

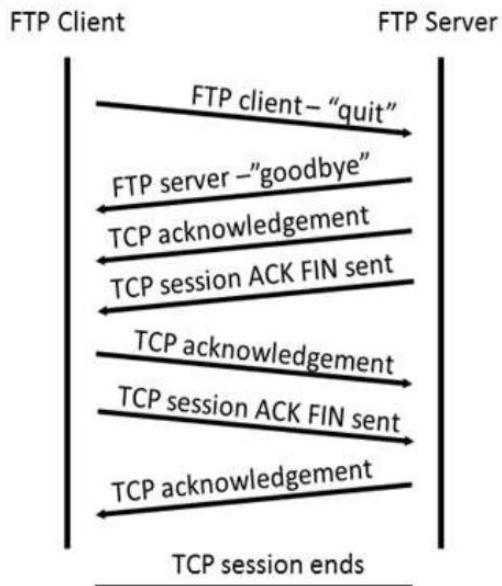


Una vez establecida una sesión TCP, puede haber tráfico FTP entre el PC de origen y el servidor FTP. El cliente y el servidor FTP se comunican entre ellos, sin saber que TCP controla y administra la sesión. Cuando el servidor FTP envía el mensaje Response: 220 (Respuesta:220) al cliente FTP, la sesión TCP en el cliente FTP envía un reconocimiento a la sesión TCP en el servidor. Esta secuencia es visible en la siguiente captura de Wireshark.



Cuando termina la sesión FTP, el cliente FTP envía un comando para “salir”. El servidor FTP reconoce la terminación de FTP con un mensaje Response: 221 Goodbye (Adiós). En este momento, la sesión TCP del servidor FTP envía un datagrama TCP al cliente FTP que anuncia la terminación de la sesión TCP. La sesión TCP del cliente FTP reconoce la recepción del datagrama de terminación y luego envía su propia terminación de sesión TCP. Cuando quien originó la

terminación TCP (servidor FTP) recibe una terminación duplicada, se envía un datagrama ACK para reconocer la terminación y se cierra la sesión TCP. Esta secuencia es visible en la captura y el diagrama siguientes.



Si se aplica un filtro **ftp**, puede examinarse la secuencia completa del tráfico FTP en Wireshark.

- k) ¿Con qué valor de dirección de correo electrónico te loqueaste en el servidor FTP anónimo? ¿Aparece dicha dirección en texto plano o cifrada?

Observe la secuencia de los eventos durante esta sesión FTP. Para recuperar el archivo "Léame", se utilizó el nombre de usuario **anonymous** (anónimo). Una vez que se completó la transferencia de archivos, el usuario finalizó la sesión FTP.

Vuelve a aplicar el filtro TCP en Wireshark para examinar la terminación de la sesión TCP. Se transmiten cuatro paquetes para la terminación de la sesión TCP. Debido a que la conexión TCP es full duplex, cada dirección debe terminar de forma independiente. Examine las direcciones de origen y destino.

57 24.719017229	192.168.22.25	198.246.117.106	FTP	72 Request: QUIT
58 24.838561886	198.246.117.106	192.168.22.25	FTP	80 Response: 221 Goodbye.
59 24.838672651	192.168.22.25	198.246.117.106	TCP	66 60068 → 21 [ACK] Seq=129 Ack=367 Win=64256 Len=130
60 24.838762818	198.246.117.106	192.168.22.25	TCP	66 21 → 60068 [FIN, ACK] Seq=367 Ack=129 Win=131072 Len=1
61 24.839639641	192.168.22.25	198.246.117.106	TCP	66 60068 → 21 [FIN, ACK] Seq=129 Ack=368 Win=64256 Len=1
62 24.958253947	198.246.117.106	192.168.22.25	TCP	66 21 → 60068 [ACK] Seq=368 Ack=130 Win=131072 Len=1

En este ejemplo, el servidor FTP no tiene más datos para enviar en la secuencia. Envía un segmento con el marcador FIN configurado en la trama 60. La MV envía un mensaje ACK para reconocer la recepción del mensaje FIN para terminar la sesión del servidor al cliente en la trama 61.

En la trama 61, el PC cliente envía un mensaje FIN al servidor FTP para terminar la sesión TCP. El servidor FTP responde con un mensaje ACK para reconocer el mensaje FIN de la MV en la trama 62. Ahora finaliza la sesión de TCP entre el servidor FTP y la MV.

6. El protocolo Telnet ha sido uno de los más utilizados para iniciar sesión remota en máquinas Linux/Unix. La desventaja de este protocolo es que los datos no se transmiten cifrados sino en texto claro, lo cual expone la información transmitida. Por otro lado, el protocolo SSH transmite sus datos cifrados de extremo a extremo.

Objetivos de la práctica:

- En esta ejercicio se utilizará la herramienta de código abierto Wireshark para capturar tráfico generado por sesiones de Telnet y SSH. Esto nos demostrará la importancia del cifrado con SSH.

PARTE 1: Examinar una sesión Telnet con WireShark.

Paso 1: Iniciar una captura de WireShark.

- a) Abra una ventana de terminal e instale un servidor de Telnet ya que este no viene instalado por defecto en la máquina virtual de Caine:

```
sudo apt update
```

```
sudo apt install telnetd
```

- b) En una ventana de terminal introduce **wireshark &** para iniciar Wireshark. Haga clic en Aceptar para continuar.
- c) En la ventana de Wireshark selecciona con doble clic, en el apartado Captura, el interfaz de loopback (**lo**) del cual se van a capturar los paquetes.

Capture



- d) Abrir otra ventana del terminal. Iniciar una sesión de Telnet al host local. Introduzca como credenciales las de la cuenta con la que accede a la máquina donde ha instalado Caine.

```

marco@marco-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
marco@marco-VirtualBox:~$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 18.04.3 LTS
marco-VirtualBox login: marco
Password: ■

```

- e) Haga clic en **Stop** (Detener) para detener la captura de Wireshark tan pronto haya introducido las credenciales del usuario.

Paso 2: Examinar la sesión de Telnet.

- f) Aplicar un filtro que solo muestre tráfico relacionado con Telnet. Ingresar **telnet** en el campo de filtro y hacer clic en **Apply**.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.006247256	127.0.0.1	127.0.0.1	TELNET	78	Telnet Data ...
6	0.006667644	127.0.0.1	127.0.0.1	TELNET	93	Telnet Data ...
8	0.006732245	127.0.0.1	127.0.0.1	TELNET	105	Telnet Data ...
9	0.006853750	127.0.0.1	127.0.0.1	TELNET	100	Telnet Data ...
10	0.007202255	127.0.0.1	127.0.0.1	TELNET	69	Telnet Data ...
11	0.007245669	127.0.0.1	127.0.0.1	TELNET	69	Telnet Data ...
12	0.007375252	127.0.0.1	127.0.0.1	TELNET	69	Telnet Data ...
13	0.007457562	127.0.0.1	127.0.0.1	TELNET	69	Telnet Data ...
14	0.007495901	127.0.0.1	127.0.0.1	TELNET	86	Telnet Data ...
16	0.050577315	127.0.0.1	127.0.0.1	TELNET	90	Telnet Data ...
18	1.305899967	127.0.0.1	127.0.0.1	TELNET	67	Telnet Data ...
19	1.306030272	127.0.0.1	127.0.0.1	TELNET	67	Telnet Data ...

- g) Haga clic en la primera de las líneas de Telnet en la sección de **Packet list** de Wireshark, y en la lista desplegable, seleccionar, Follow > TCP Stream.

4	0.006247256	127.0.0.1	127.0.0.1	TELNET	78	Telnet Data ...
6	0.006667644	127.0.0.1	127.0.0.1	Mark/Unmark Packet	Control+M	ata ...
8	0.006732245	127.0.0.1	127.0.0.1	Ignore/Unignore Packet	Control+D	ata ...
9	0.006853750	127.0.0.1	127.0.0.1	Set/Unset Time Reference	Control+T	ata ...
10	0.007202255	127.0.0.1	127.0.0.1	Time Shift...	Control+Mayúsculas+T	ata ...
11	0.007245669	127.0.0.1	127.0.0.1	Packet Comment...	Control+Alt+C	ata ...
12	0.007375252	127.0.0.1	127.0.0.1	Edit Resolved Name		ata ...
13	0.007457562	127.0.0.1	127.0.0.1	Apply as Filter		ata ...
14	0.007495901	127.0.0.1	127.0.0.1	Prepare a Filter		ata ...
16	0.050577315	127.0.0.1	127.0.0.1	Conversation Filter		ata ...
18	1.305899967	127.0.0.1	127.0.0.1	Colorize Conversation		ata ...
19	1.306030272	127.0.0.1	127.0.0.1	SCTP		ata ...
21	1.459090051	127.0.0.1	127.0.0.1	Follow		TCP Stream Control+Alt+Mayúsculas+T
22	1.459297506	127.0.0.1	127.0.0.1			ata ...
24	1.732719057	127.0.0.1	127.0.0.1			ata ...
25	1.732774580	127.0.0.1	127.0.0.1			ata ...
27	1.957645310	127.0.0.1	127.0.0.1			ata ...
28	1.957712981	127.0.0.1	127.0.0.1			ata ...
30	2.039729525	127.0.0.1	127.0.0.1			ata ...
31	2.039804989	127.0.0.1	127.0.0.1			ata ...
33	2.457234926	127.0.0.1	127.0.0.1			ata ...
34	2.457799993	127.0.0.1	127.0.0.1			ata ...

- h) En la ventana **Follow TCP Stream** se muestran los datos para su sesión de Telnet con la Máquina Virtual. Toda la sesión se muestra como texto plano, incluida la contraseña. Observar que el nombre de usuario que se introdujo aparece con

caracteres duplicados. Esto se debe al ajuste de echo en Telnet para permitirle ver los caracteres que escribe en la pantalla.

```
Wireshark · Follow TCP Stream (tcp.stream eq 0) · Loopback: lo
.... .#...'. .... ..!"'.....#.....!...".....
....#.....'. ....P..... 38400,38400....#.marco-VirtualBox:
0....'.DISPLAY.marco-VirtualBox:0....xterm.....Ubuntu 18.04.3 LTS
marco-VirtualBox login: mmaarrccoo
.
Password: practicas2021
.
Last login: Fri Jul  8 18:19:30 CEST 2022 from localhost on pts/1
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-32-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

Pueden actualizarse 3 paquetes.
0 actualizaciones son de seguridad.

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
.]0;marco@marco-VirtualBox: ~.marco@marco-VirtualBox:~$
```

- i) Cuando termine de revisar la sesión de Telnet en la ventana Follow TCP Stream, hacer clic en Close (Cerrar). Escribir exit en el terminal para salir de la sesión de Telnet.

PARTE 2: Examinar una sesión SSH con Wireshark.

- j) En primer lugar, inicie el servicio sshd (el cual viene preinstalado con CAINE) si no se encuentra ya iniciado. Para arrancar el servicio, introduzca el siguiente comando:

```
sudo systemctl start sshd.service
```

- k) Iniciar una captura de Wireshark en la interfaz Loopback: **lo**
- l) Establezca una sesión de SSH con el host local. En el prompt del terminal, introducir **ssh localhost**. Responda yes para seguir con la conexión. Inicie sesión con la cuenta de usuario con la que inició sesión en CAINE.

```
marco@marco-VirtualBox:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:SxjkWIVCmheYlhPSQ4rfDy6/0007Y9VTqKyVL2tvyQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
marco@localhost's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-32-generic x86_64)
```

- m) Aplicar un filtro de SSH a los datos de la captura de Wireshark. Introducir **ssh** en el campo de filtro y haga clic en Aplicar.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.037769480	127.0.0.1	127.0.0.1	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubu
9	0.040536619	127.0.0.1	127.0.0.1	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubu
11	0.040758936	127.0.0.1	127.0.0.1	SSHv2	1426	Client: Key Exchange Init
12	0.041743775	127.0.0.1	127.0.0.1	SSHv2	1146	Server: Key Exchange Init
13	0.044342000	127.0.0.1	127.0.0.1	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
14	0.049424736	127.0.0.1	127.0.0.1	SSHv2	518	Server: Diffie-Hellman Key Exchange Reply, New Keys
30	3.123783370	127.0.0.1	127.0.0.1	SSHv2	82	Client: New Keys
32	3.164447975	127.0.0.1	127.0.0.1	SSHv2	110	Client: Encrypted packet (len=44)
34	3.164538743	127.0.0.1	127.0.0.1	SSHv2	110	Server: Encrypted packet (len=44)
36	3.164616259	127.0.0.1	127.0.0.1	SSHv2	134	Client: Encrypted packet (len=68)
37	3.165484878	127.0.0.1	127.0.0.1	SSHv2	118	Server: Encrypted packet (len=52)
69	9.272691747	127.0.0.1	127.0.0.1	SSHv2	214	Client: Encrypted packet (len=148)
72	9.281043610	127.0.0.1	127.0.0.1	SSHv2	94	Server: Encrypted packet (len=28)
74	9.281137871	127.0.0.1	127.0.0.1	SSHv2	178	Client: Encrypted packet (len=112)
78	9.701715697	127.0.0.1	127.0.0.1	SSHv2	566	Server: Encrypted packet (len=500)
80	9.744562453	127.0.0.1	127.0.0.1	SSHv2	110	Server: Encrypted packet (len=44)
82	9.744671269	127.0.0.1	127.0.0.1	SSHv2	1146	Client: Encrypted packet (len=1080)
84	9.745762330	127.0.0.1	127.0.0.1	SSHv2	174	Server: Encrypted packet (len=108)
85	9.751819245	127.0.0.1	127.0.0.1	SSHv2	678	Server: Encrypted packet (len=612)
89	9.918745633	127.0.0.1	127.0.0.1	SSHv2	158	Server: Encrypted packet (len=92)

- n) Haz clic en la primera de las líneas de SSH en la sección de **Packet list** de Wireshark, y en la lista desplegable, seleccionar, **Follow > TCP Stream**.
- o) Examinar la ventana **Follow TCP Stream** en la sesión de SSH. Los datos se cifraron y son ilegibles. Comparar los datos de la sesión de SSH con los datos de la sesión de Telnet.
- p) Para finalizar cierre la sesión de ssh con exit y cierre Wireshark.
7. Los ataques de inyección SQL permiten que los hackers maliciosos escriban sentencias SQL en un sitio web y reciban una respuesta de la base de datos. Esto permite a los atacantes manipular los datos actuales de la base de datos, suplantar identidades y modificar o destruir información.

Objetivos de la práctica:

- En este ejercicio examinaremos un archivo PCAP para que veamos un ataque anterior a una base de datos SQL.

Paso 1: Abrir Wireshark y cargar el archivo PCAP

- En una ventana de terminal introduce **wireshark &** para iniciar Wireshark. Haga clic en Aceptar para continuar.
- Descargue del Campus Virtual de la asignatura el archivo denominado **Recursos Prácticas->Práctica 5->SQL_Lab.pcap**.
- Abra el archivo anterior dentro de Wireshark para mostrar el tráfico de red capturado.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.4	10.0.2.15	TCP	74	35614 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1464
2	0.000315	10.0.2.15	10.0.2.4	TCP	74	80 → 35614 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
3	0.000349	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0
4	0.000681	10.0.2.4	10.0.2.15	HTTP	654	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	0.002149	10.0.2.15	10.0.2.4	TCP	66	80 → 35614 [ACK] Seq=1 Ack=589 Win=30208 Len=0
6	0.005700	10.0.2.15	10.0.2.4	HTTP	430	HTTP/1.1 302 Found
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=589 Ack=365 Win=30336 Len=0
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	GET /dvwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1019 Ack=3406 Win=36480 Len=0
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80 → 35638 [ACK] Seq=1 Ack=471 Win=235 Len=0
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80 → 35640 [ACK] Seq=1 Ack=512 Win=235 Len=0
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80 → 35642 [ACK] Seq=1 Ack=565 Win=236 Len=0
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1

- d) ¿A cuánto tiempo (en segundos) corresponde la captura?
- e) ¿Cuántos paquetes fueron capturados?
- f) En función de la información que proporcionan los paquetes capturados, ¿cuáles son las dos direcciones involucradas en este ataque de inyección SQL?

Paso 2: Ver el ataque de inyección SQL.

- g) Dentro de la captura de Wireshark, haga clic derecho en la línea 13 y seleccione **Follow > HTTP Stream**. Se eligió la línea 13 porque es una solicitud GET HTTP. Esto será muy útil para seguir el flujo de datos a medida que lo ven las capas de aplicación y se genera una prueba de consulta para la inyección SQL.

```

GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.15/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=ml2n7d0t4rem6k0n4is82u5157
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Mon, 06 Feb 2017 14:18:22 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 1443
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
    </head>
    <body>
        <h1>Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</h1>
        <p>SQL injection vulnerability found!</p>
        <pre>SELECT * FROM users WHERE id = 1 AND password = '123456'</pre>
        <form action="" method="POST">
            <input type="text" name="id" value="1" />
            <input type="submit" value="Submit" />
        </form>
    </body>
</html>

```

Entire conversation (5.894 bytes) Show and save data as ASCII

Find: Find Next

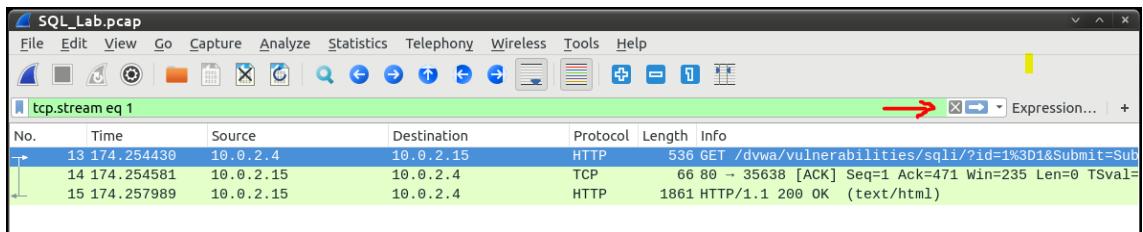
Ayuda Filter Out This Stream Print Save as... Back Cerrar

El tráfico de origen se muestra en rojo. El origen ha enviado una solicitud GET al host **10.0.2.15**. En color azul, el dispositivo de destino le está respondiendo al origen.

- h) En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.

El atacante ha ingresado una consulta (**1=1**) en un cuadro de búsqueda de UserID en el objetivo 10.0.2.15 para ver si la aplicación es vulnerable a la inyección SQL. En lugar de responder con un mensaje de error de inicio de sesión, la aplicación respondió con un registro (record) de la base de datos. El atacante ha verificado que puede ingresar un comando SQL y que la base de datos le responderá. El string de búsqueda "**1=1**" crea una sentencia SQL que siempre será verdadera. En el ejemplo no importa lo que se haya ingresado en el campo, siempre será verdadera.

Cierra la ventana **Follow HTTP Stream**. Haga clic en **Clear display filter** para mostrar la conversación completa de Wireshark.



Paso 3: El ataque de inyección SQL continua.

- i) Dentro de la captura de Wireshark, haga clic derecho en la línea 19, y luego haga clic en **Follow > HTTP Stream**. En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.

El atacante ha ingresado una consulta (**1' or 1=1 union select database(), user()#**) en un cuadro de búsqueda de UserID en el objetivo **10.0.2.15**. En lugar de responder con un mensaje de error de inicio de sesión (login failure), la aplicación respondió con la siguiente información:

```
<pre>ID: 1' or 1=1 union select database(), user()#<br />First
name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select database(),
user()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union
select database(), user()#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1'
or 1=1 union select database(), user()#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select database(), user()#<br />First name:
Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select database(),
user()#<br />First name: dvwa<br />Surname: root@localhost</pre>
```

- j) Indique cuántas cuentas de usuario se han descubierto.

Cierra la ventana **Follow HTTP Stream**. Haga clic en **Clear display filter** para mostrar la conversación completa de Wireshark.

Paso 4: El ataque de inyección SQL proporciona información del sistema.

- k) Dentro de la captura de Wireshark, haga clic derecho en la línea **22** y seleccione **Follow > HTTP Stream**. El tráfico de origen se muestra en rojo, y está enviando la solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.
- l) En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.

Wireshark · Follow HTTP Stream (tcp.stream eq 4) · SQL_Lab.pcap

```

<li onclick="window.location='../../about.php'" class=""><a href="../../about.php">About</a></li>
</ul><ul class="menuBlocks"><li onclick="window.location='../../logout.php'" class=""><a href="../../logout.php">Logout</a></li>
</ul>
</div>
</div>
<div id="main_body">
<div class="body_padded">
<h1>Vulnerability: SQL Injection</h1>
<div class="vulnerable_code_area">
<form action="#" method="GET">
<p>
User ID:
<input type="text" size="15" name="id">
<input type="submit" name="Submit" value="Submit">
</p>
</form>
<pre>ID: 1' or 1=1 union select null, version ()#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: <br />Surname: 5.7.12-0ubuntu1.1</pre>
</div>

```

1 client pkt(s), 1 server pkt(s), 1 turn(s).

Entire conversation (6.548 bytes) Show and save data as ASCII

Find: 1=1 Find Next

Ayuda Filter Out This Stream Print Save as... Back Cerrar

- m) El atacante ha ingresado una consulta (**1' or 1=1 union select null, version ()#**) en un cuadro de búsqueda de UserID en el objetivo 10.0.2.15 para localizar el identificador de la versión. Observe que el identificador de versión se encuentra al final del resultado justo antes del cierre del código HTML </pre></div>. ¿Cuál es la versión?

Cierra la ventana **Follow HTTP Stream**. Haga clic en **Clear display filter** para mostrar la conversación completa de Wireshark.

Paso 5: El atacante trata de encontrar las tablas que hay en la Base de Datos

- n) Dentro de la captura de Wireshark, haga clic derecho en la línea **25**, y luego seleccione **Follow > HTTP Stream**. Se ha enviado una solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen. En el apartado **Find**, escriba **users**. Haga clic en **Find Next**.

El atacante ha ingresado una consulta (**1' or 1=1 union select null, table_name from information_schema.tables#**) en un cuadro de búsqueda de UserID en el objetivo 10.0.2.15 para ver todas las tablas de la base de datos. Esto proporciona una enorme salida de muchas tablas, ya que el atacante especificó “null” sin más especificaciones.

- o) ¿Qué haría el comando modificado por el atacante: (**1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users'**)?

Cierra la ventana **Follow HTTP Stream**. Haga clic en Clear display filter para mostrar la conversación completa de Wireshark.

Paso 6: El ataque de inyección SQL concluye.

- p) Dentro de la captura de Wireshark, haga clic derecho en la línea **28** y luego seleccione **Follow > HTTP Stream**. El origen se muestra en rojo. Se ha enviado una solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.
- q) Haga clic en **Find** y escriba **1=1**. Después, busque la entrada "1=1". Después de encontrarla, haga clic en Cancel en el cuadro de búsqueda de texto "Find".

El atacante ha ingresado una consulta (**1'or 1=1 union select user, password from users#**) en un cuadro de búsqueda de UserID en el objetivo 10.0.2.15 para obtener nombres de usuario y hashes de contraseñas.

- r) ¿Qué usuario tiene "**8d3533d75ae2c3966d7e0d4fcc69216b**" como hash de su contraseña?
 - s) Utilice un sitio web como <https://crackstation.net/> para copiar el hash de la contraseña en el decodificador de hashes de contraseñas y comenzar a decodificarlo. ¿Cuál es la contraseña en texto plano?
8. En este ejercicio, extraeremos un fichero de un flujo de datos capturado en una sesión de Wireshark y analizaremos dicho fichero.

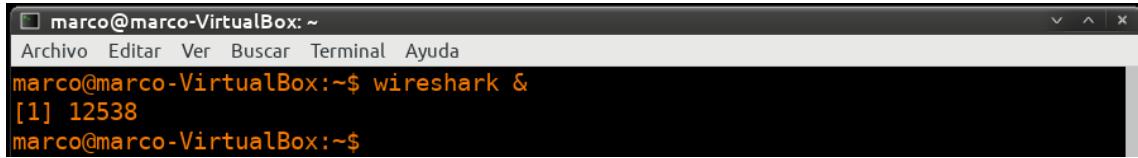
Objetivos de la práctica:

- En este ejercicio se analizará el tráfico de un archivo pcap previamente capturado y extraeremos un fichero “ejecutable” del archivo de captura de paquetes.

PARTE 1: Analizar archivos de tráfico precapturados.

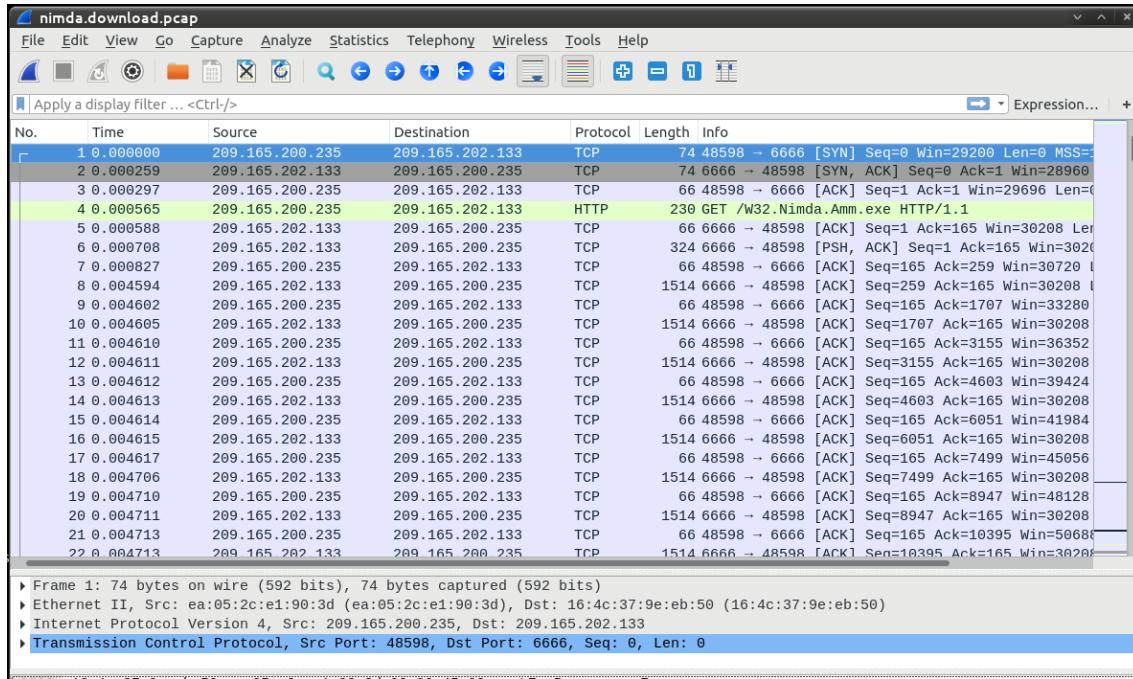
- a) Descarga del Campus Virtual el fichero denominado **Recursos Prácticas->Práctica 5->nimda.captura.pcap**.

- b) En una ventana de terminal introduce **wireshark &** para iniciar Wireshark.
 Haga clic en Aceptar para continuar.

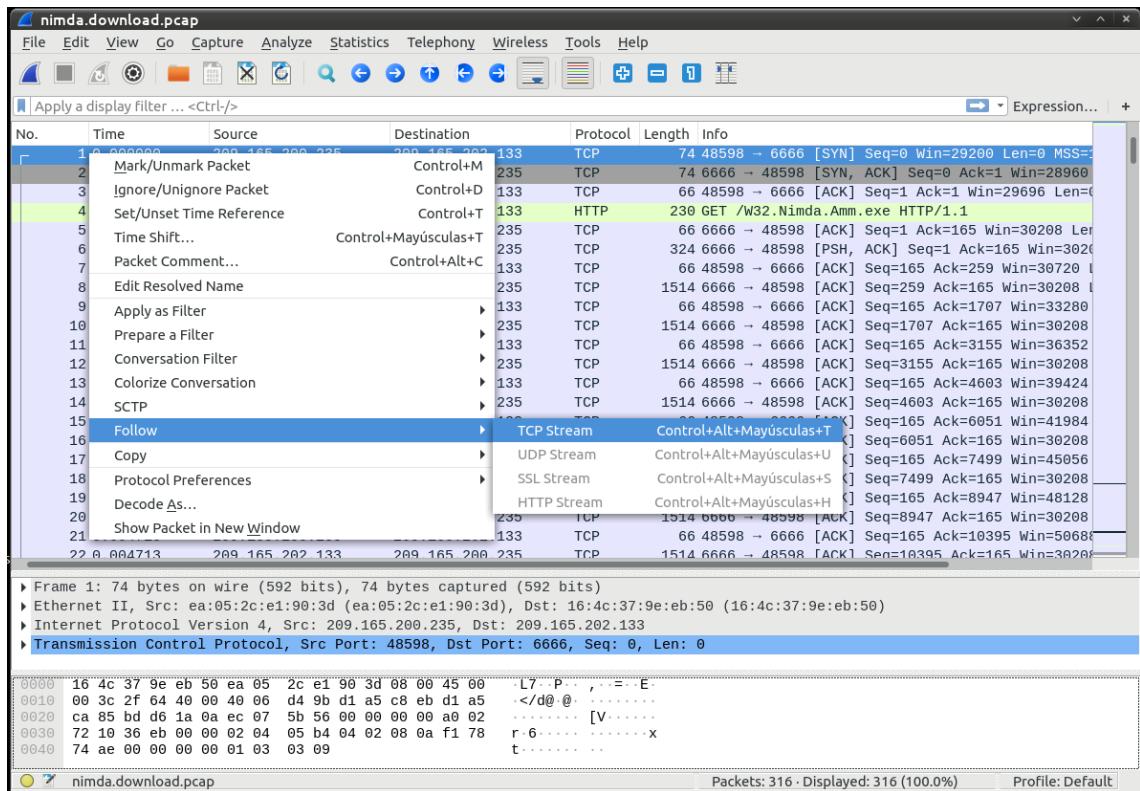


```
marco@marco-VirtualBox:~$ wireshark &
[1] 12538
marco@marco-VirtualBox:~$
```

- c) Abra el archivo anterior dentro de Wireshark para mostrar el tráfico de red capturado.



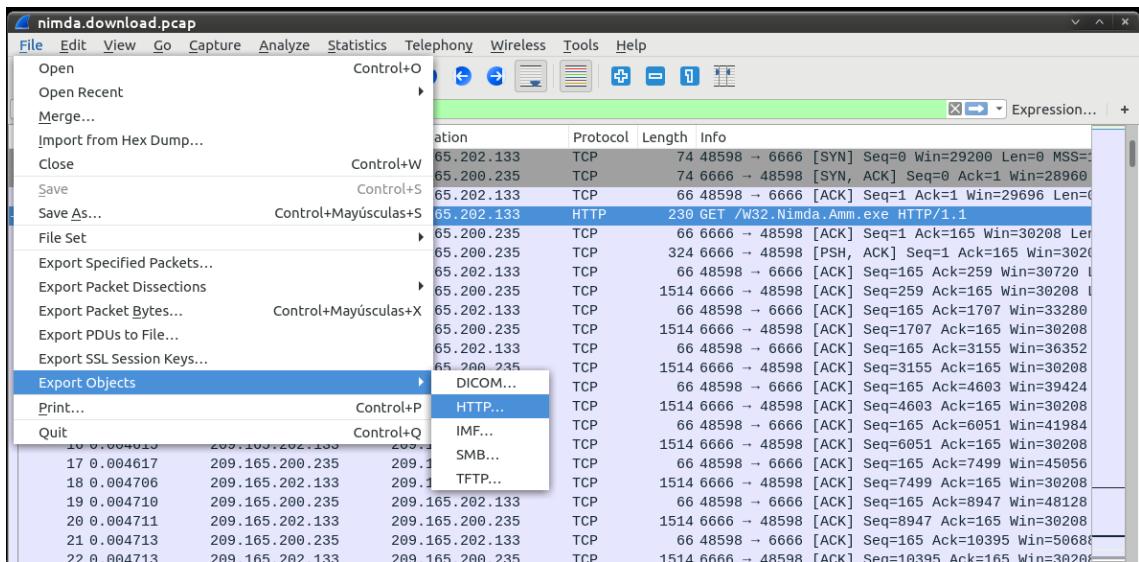
- d) Los paquetes del uno al tres son el protocolo de enlace TCP. En el cuarto paquete se muestra la solicitud correspondiente al archivo de malware. A modo de confirmación de lo que ya se sabía, la solicitud se realizó por HTTP, y se envió como solicitud GET. Como HTTP se ejecuta por TCP, se puede utilizar la característica Follow TCP Stream (Seguir flujo de TCP) de Wireshark para reconstruir la transacción TCP. Seleccione el primer paquete TCP de la captura, es un paquete SYN. Haz clic derecho y elige Follow >TCP Stream.



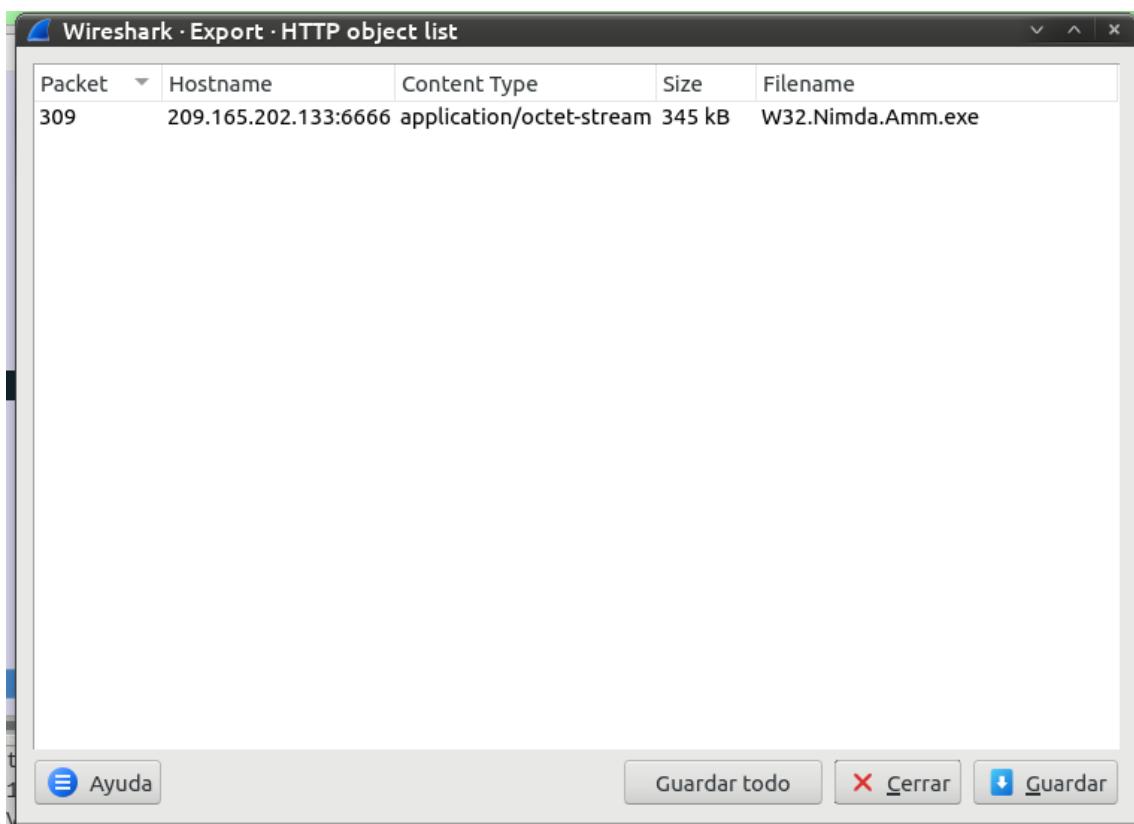
- e) En Wireshark se abre otra ventana con los detalles correspondientes a todo el flujo de TCP.
- f) ¿Qué son todos esos símbolos que se ven en la ventana de Follow TCP Stream?
- g) En la ventana de Follow TCP Stream, haga clic en **Close** (Cerrar) para regresar al archivo **nimda.download.pcap** de Wireshark.

PARTE 2: Extraer archivos descargados desde archivos PCAP.

- h) En ese cuarto paquete del archivo **nimda.download.pcap**, observa que la solicitud HTTP GET se generó desde **209.165.200.235** hacia **209.165.202.133**. En la columna **Info** (Información) también se ve que de hecho se trata de la solicitud GET correspondiente al archivo.
- i) Con el paquete de la solicitud GET seleccionado, diríjase a **File > Export Objects HTTP** desde el menú de Wireshark.



- j) En Wireshark se mostrarán todos los objetos HTTP presentes en el flujo TCP que contiene la solicitud GET. En este caso, el único archivo presente en la captura es **W32.Nimda.Amm.exe**.



- k) En la ventana **HTTP object list** (Lista de objetos HTTP), seleccione el archivo **W32.Nimda.Amm.exe** y haga clic en **Save As** (Guardar como), en la parte inferior de la pantalla.
- l) Abra una terminal, vaya a la carpeta donde guardó el archivo y compruebe con el comando file el tipo de archivo que es.

m) ¿Cuál sería el siguiente paso para comprobar si el archivo es realmente un Malware?