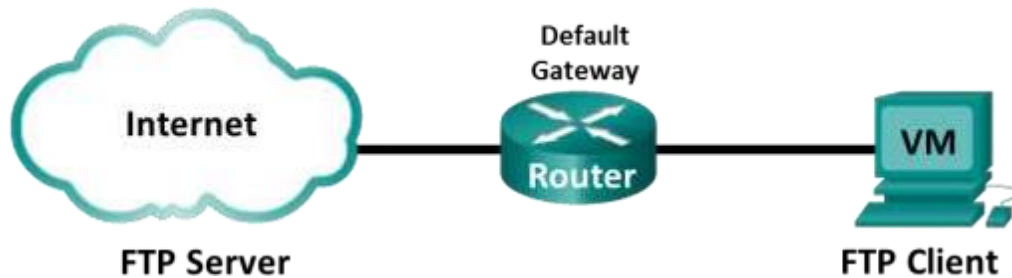


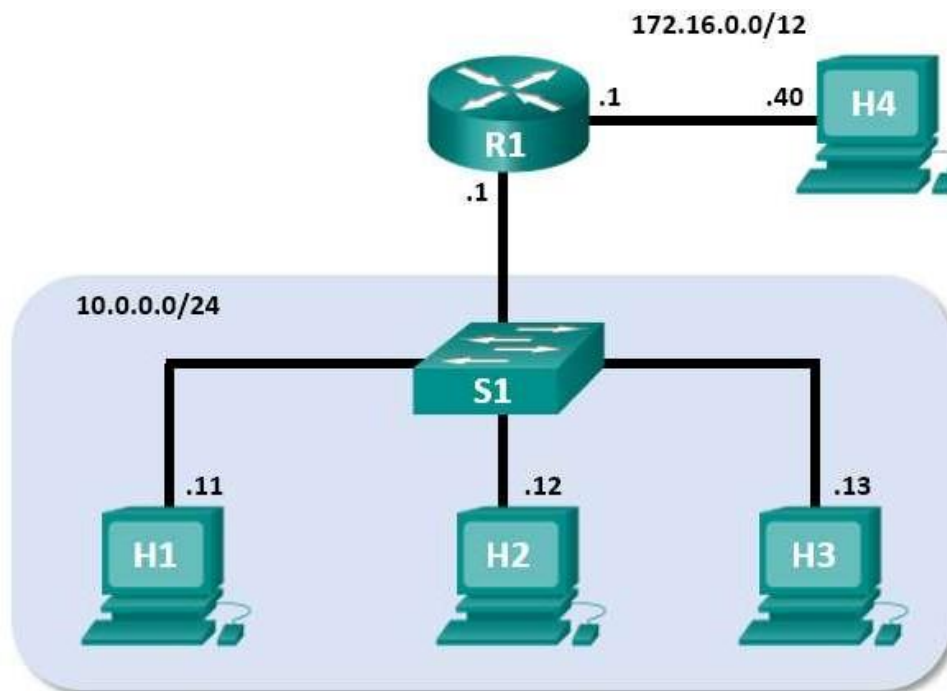
# Práctica de laboratorio: Utilizar Wireshark para examinar capturas de TCP y UDP

## Topología - Parte 1 (FTP)



La parte 1 destacará una captura de TCP de una sesión FTP. Esta topología consiste en la Máquina Virtual de la estación de trabajo (Workstation VM) de CyberOps con acceso a Internet.

## Topología de mini red - Parte 2 (TFTP)



## Objetivos

Parte 1: Identificar campos de encabezado y operación TCP mediante una captura de sesión FTP de Wireshark.

Parte 2: Identificar campos de encabezado y operación UDP mediante una captura de sesión TFTP de Wireshark.

### Aspectos básicos/situación

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 FTP, entre otros. UDP proporciona soporte de capa de transporte para el sistema de nombres de dominio (DNS) y TFTP, entre otros.

En la Parte 1 de esta práctica de laboratorio utilizarán la herramienta de código abierto 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 utiliza la línea de comandos del terminal para establecer una conexión a un servidor FTP anónimo y descargar un archivo. En la Parte 2 de esta práctica de laboratorio, utilizará Wireshark para capturar y analizar campos de encabezado UDP correspondientes a transferencias de archivos TFTP entre computadoras host de Mininet.

### Recursos necesarios

- VM CyberOps Workstation
- Acceso a Internet

### Instrucciones

#### Parte 1: Identificar campos de encabezado y operación TCP mediante una captura de sesión FTP de Wireshark

En la parte 1, utilizará Wireshark para capturar una sesión FTP e inspeccionar los campos de encabezado de TCP.

##### Paso 1: Iniciar una captura de Wireshark

- Abra la VM CyberOps Workstation e inicien sesión. Abra una ventana del terminal e inicie Wireshark. El ampersand (&) envía el proceso a segundo plano y le permite continuar trabajando en el mismo terminal.

```
[analyst@secOps ~]$ wireshark &
```

- Inicie una captura de Wireshark correspondiente a la interfaz **enp0s3**.

- 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.

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

##### Paso 2: Descargar el archivo Readme.

- Localicen y descarguen el archivo Readme; para ello, introduzcan el comando **ls** para generar una lista de los archivos.

```
ftp> ls
```

```
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 Feb 2 11:21 pub
-rwxrwxrwx 1 owner group 1428 May 13 1999 Readme
-rwxrwxrwx 1 owner group 383 May 13 1999 Siteinfo
-rwxrwxrwx 1 owner group 0 May 17 2005 up.htm
drwxrwxrwx 1 owner group 0 May 20 2010 w3c
-rwxrwxrwx 1 owner group 202 Sep 22 1998 welcome.msg
226 Transfer complete.
```

**Nota:** Debe recibir los siguientes mensajes:

```
421 Service not available, remote server has closed connection
ftp: No control connection for command
```

```
501 Server cannot access argument
500 command not understood
ftp: bind: Address already in use
```

Si esto sucede, significa que en ese momento el servidor FTP no funciona. Sin embargo, puede proceder con el resto del laboratorio analizando los paquetes que fue capaz de capturar y leyendo los paquetes que no capturó. También puede volver a la práctica de laboratorio más tarde para ver si el servidor FTP volvió a funcionar.

- b. Introduzca el comando **get Readme** para descargar el archivo. Cuando finalice la descarga, introduzca el comando **quit** para salir. (**Nota:** Si no puede descargar el archivo, puede proceder con el resto del laboratorio.)

```
ftp> get 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 0.056 seconds (24.9 kbytes/s)
```

- c. Una vez finalizada la transferencia, introduzca **quit** para salir de ftp.

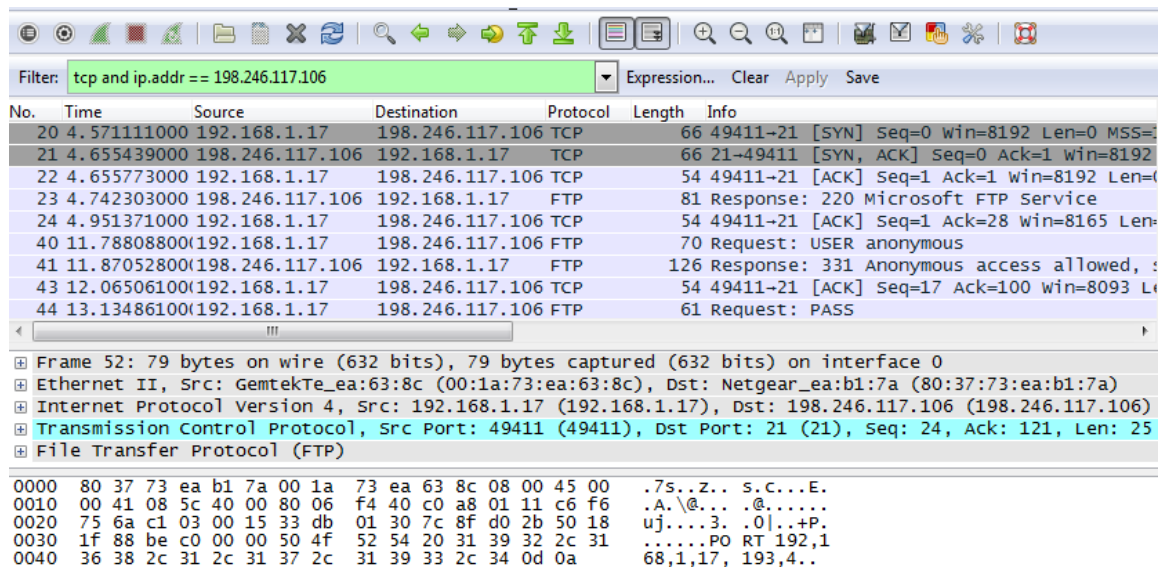
### Paso 3: Detener la captura Wireshark

### Paso 4: Ver la ventana principal de Wireshark

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](http://ftp.cdc.gov) cuando se creó esta práctica de laboratorio. Sus direcciones IP pueden ser diferentes. Si es así, busquen el primer paquete TCP

que inició el Protocolo de enlace de 3 vías con [ftp.cdc.gov](http://ftp.cdc.gov). La dirección IP de destino es la dirección IP que deben utilizar para el filtro.



The screenshot shows the Wireshark interface with a packet capture filter applied: `tcp and ip.addr == 198.246.117.106`. The packet list shows several TCP and FTP packets. The selected packet (No. 21) is a SYN, ACK packet from 198.246.117.106 to 192.168.1.17. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol (TCP) layers. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
20	4.571111000	192.168.1.17	198.246.117.106	TCP	66	49411→21 [SYN] Seq=0 win=8192 Len=0 MSS=
21	4.655439000	198.246.117.106	192.168.1.17	TCP	66	21→49411 [SYN, ACK] Seq=0 Ack=1 win=8192
22	4.655773000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=1 Ack=1 win=8192 Len=0
23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
24	4.951371000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=1 Ack=28 win=8165 Len=0
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, s
43	12.065061000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=17 Ack=100 win=8093 L
44	13.134861000	192.168.1.17	198.246.117.106	FTP	61	Request: PASS

Frame 52: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0  
Ethernet II, Src: GemtekTe\_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear\_ea:b1:7a (80:37:73:ea:b1:7a)  
Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)  
Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 24, Ack: 121, Len: 25  
File Transfer Protocol (FTP)

0000 80 37 73 ea b1 7a 00 1a 73 ea 63 8c 08 00 45 00 .7s..z..S.C...E.  
0010 00 41 08 5c 40 00 80 06 f4 40 c0 a8 01 11 c6 f6 .A.\@...@.....  
0020 75 6a c1 03 00 15 33 db 01 30 7c 8f d0 2b 50 18 uj....3..0|..+P.  
0030 1f 88 be c0 00 00 50 4f 52 54 20 31 39 32 2c 31 .....PO RT 192,1  
0040 36 38 2c 31 2c 31 37 2c 31 39 33 2c 34 0d 0a 68,1,17, 193,4..

**Nota:** La interfaz del Wireshark puede verse ligeramente diferente a la imagen de arriba.

### Paso 5: Analizar los campos 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.

20	4.571111000	192.168.1.17	198.246.117.106	TCP	66	49411→21 [SYN] Seq=0 win=8192 Len=0 MSS=
21	4.655439000	198.246.117.106	192.168.1.17	TCP	66	21→49411 [SYN, ACK] Seq=0 Ack=1 win=8192
22	4.655773000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=1 Ack=1 win=8192 Len=0

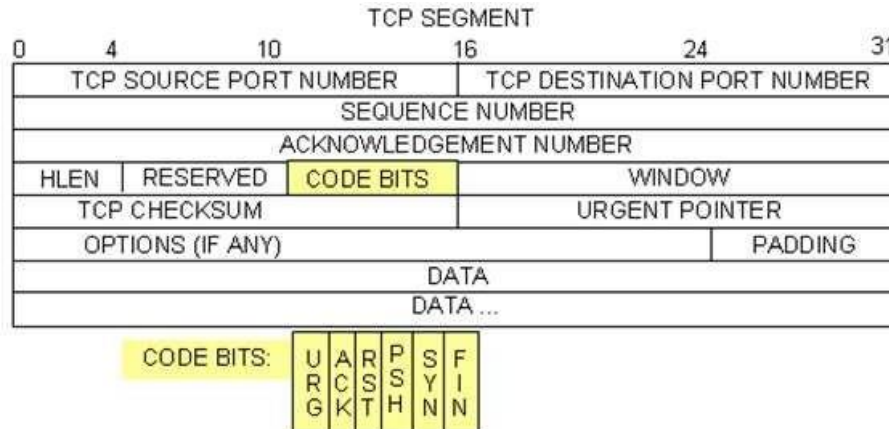
TCP se utiliza en forma continua durante una sesión para controlar la entrega de datagramas, verificar la llegada de datagramas y administrar el tamaño de la ventana. Para cada intercambio de datos entre el cliente FTP y el servidor FTP, se inicia una nueva sesión TCP. Al término de la transferencia 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 los primeros datagramas TCP del host, y expanda las porciones del datagrama de TCP, como se muestra a continuación.

```

Frame 20: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 0, Len: 0
  Source Port: 49411 (49411)
  Destination Port: 21 (21)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
  Header Length: 32 bytes
  ... 0000 0000 0010 = Flags: 0x002 (SYN)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion window Reduced (cwr): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...0 = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... ..... 0.. = Reset: Not set
    .... .... .1. = Syn: Set
    .... .... ...0 = Fin: Not set
  window size value: 8192
  [Calculated window size: 8192]
  Checksum: 0x5bba [validation disabled]
  Urgent pointer: 0
  Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No-0
  
```

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.

**Nota:** En la captura anterior de Wireshark, el puerto de destino es 21, que es el FTP. Los servidores FTP escuchan las conexiones de cliente FTP en el puerto 21.

- **Sequence number** (Número de secuencia) especifica el número del último octeto en un segmento.
- **Acknowledgment number** (Número de reconocimiento) especifica el siguiente octeto 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).

Utilice 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 VM al servidor CDC (solamente el bit SYN está definido en 1):

Descripción	Resultados del Wireshark
Dirección IP de origen	
Dirección IP de destino	
Número de puerto de origen	
Número de 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 de CDC confirma que recibió la solicitud de la VM. Observe los valores de los bits de SYN y ACK.

```

+ Frame 21: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
+ Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)
+ Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)
- Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 0, Ack: 1, Len: 0
  Source Port: 21 (21)
  Destination Port: 49411 (49411)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 32 bytes
  - ... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    + .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
  window size value: 8192
  [Calculated window size: 8192]
  + Checksum: 0x0ee7 [validation disabled]
  Urgent pointer: 0
  + Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No
  + [SEQ/ACK analysis]
  
```

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

Descripción	Resultados del Wireshark
Dirección IP de origen	
Dirección IP de destino	
Número de puerto de origen	
Número de puerto de destino	
Número de secuencia	
Número de acuse de recibo	
Longitud del encabezado	
Tamaño de la ventana	

## Práctica de laboratorio: Utilizar Wireshark para examinar capturas de TCP y UDP

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.

```
⊞ Frame 22: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
⊞ Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
⊞ Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 1, Ack: 1, Len: 0
    Source Port: 49411 (49411)
    Destination Port: 21 (21)
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 1 (relative sequence number)
    Acknowledgment number: 1 (relative ack number)
    Header Length: 20 bytes
    ⊞ ... 0000 0001 0000 = Flags: 0x010 (ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0... .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
    window size value: 8192
    [Calculated window size: 8192]
    [window size scaling factor: 1]
    ⊞ Checksum: 0x4f6a [validation disabled]
    Urgent pointer: 0
    ⊞ [SEQ/ACK analysis]
```

Complete la siguiente información sobre el mensaje de ACK.

Descripción	Resultados del Wireshark
Dirección IP de origen	
Dirección IP de destino	
Número de puerto de origen	
Número de puerto de destino	
Número de secuencia	
Número de acuse de recibo	
Longitud del encabezado	
Tamaño de la ventana	

¿Cuántos otros datagramas TCP contenían un bit SYN?

Una vez establecida una sesión TCP, puede haber tráfico FTP entre la PC 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

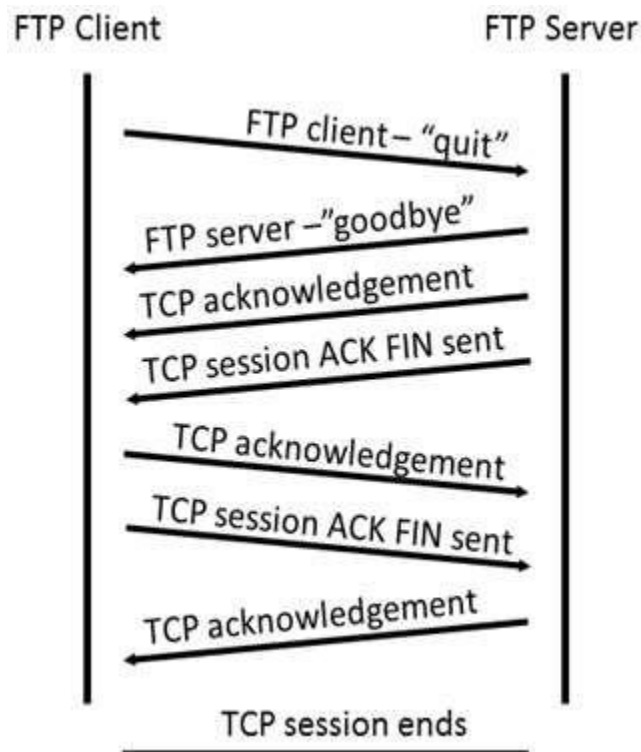


un reconocimiento a la sesión TCP en el servidor. Esta secuencia es visible en la siguiente captura de Wireshark.

23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
24	4.951371000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=1 Ack=28 win=8165 Len=
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, :

Frame 23: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0  
 Ethernet II, Src: Netgear\_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe\_ea:63:8c (00:1a:73:ea:63:8c)  
 Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)  
 Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 1, Ack: 1, Len: 27  
 File Transfer Protocol (FTP)  
 220 Microsoft FTP Service\r\n  
 Response code: Service ready for new user (220)  
 Response arg: Microsoft FTP Service

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. 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.

No.	Time	Source	Destination	Protocol	Length	Info
23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, send the command
44	13.134861000	192.168.1.17	198.246.117.106	FTP	61	Request: PASS
46	13.328294000	198.246.117.106	192.168.1.17	FTP	75	Response: 230 User logged in.
51	16.352248000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,4
52	16.682680000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168,1,17,193,4
54	17.354538000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT command successful
55	17.363442000	192.168.1.17	198.246.117.106	FTP	60	Request: NLST
56	17.442635000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 opening ASCII mode data connection
62	19.897441000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
73	24.297181000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,5
75	24.607498000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168,1,17,193,5
82	25.136886000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT command successful
83	25.142329000	192.168.1.17	198.246.117.106	FTP	67	Request: RETR Readme
101	25.270185000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 opening ASCII mode data connection
127	27.784523000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.

Vuelva 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. Porque la conexión TCP es full duplex, cada dirección debe terminar de forma independiente. Examine las direcciones de origen y destino.

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 149. La PC envía un mensaje ACK para reconocer la recepción del mensaje FIN para terminar la sesión del servidor al cliente en la trama 150.

En la trama 151, la PC 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 PC en la trama 152. Ahora finaliza la sesión de TCP entre el servidor FTP y la PC.

No.	Time	Source	Destination	Protocol	Length	Info
147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.
149	30.566467000	198.246.117.106	192.168.1.17	TCP	54	21->49411 [FIN, ACK] Seq=325 Ack=99 win=1
150	30.566532000	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=99 Ack=326 win=7868
151	30.566799000	192.168.1.17	198.246.117.106	TCP	54	49411->21 [FIN, ACK] Seq=99 Ack=326 win=7868
152	30.667770000	198.246.117.106	192.168.1.17	TCP	54	21->49411 [ACK] Seq=326 Ack=100 win=132096

## Parte 2: Identificar campos de encabezado y operación UDP mediante una captura de sesión TFTP de Wireshark

En la parte 2, utilizará Wireshark para capturar una sesión TFTP e inspeccionar los campos de encabezado de UDP.

### Paso 1: Iniciar Mininet y el servicio tftpd.

- Inicien Mininet. Introduzca **cyberops** como la contraseña cuando se los solicite el sistema.

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

```
[sudo] contraseña para analyst:
```

- Inicien H1 y H2 en el cursor **mininet>**.

```
*** Starting CLI:
```

```
mininet> xterm H1 H2
```

- c. En la ventana del terminal de **H1**, inicie el servidor tftpd con el script provisto.

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/start_tftpd.sh
[root@secOps analyst]#
```

### Paso 2: Crear un archivo para la transferencia tftp

- a. Cree un archivo de texto en el cursor del terminal de **H1**, en la carpeta /srv/tftp/.

```
[root@secOps analyst]# echo "This file contains my tftp data." >
/srv/tftp/my_tftp_data
```

- b. Verifique que se haya creado el archivo con los datos deseados en la carpeta.

```
[root@secOps analyst]# cat /srv/tftp/my_tftp_data
This file contains my tftp data.
```

- c. Debido a la medida de seguridad correspondiente a este servidor tftp en particular, el nombre del archivo receptor ya tiene que existir. En **H2**, creen un archivo de nombre **my\_tftp\_data**.

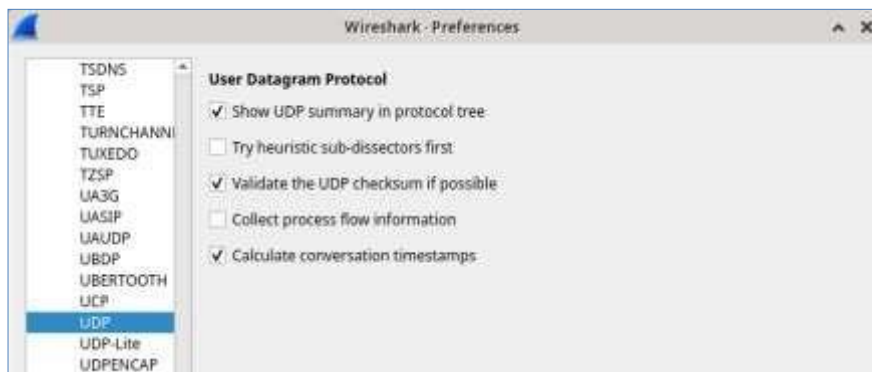
```
[root@secOps analyst]# touch my_tftp_data
```

### Paso 3: Capturar una sesión de TFTP en Wireshark

- a. Inicie Wireshark en **H1**.

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

- b. En el menú **Editar**, elija **Preferencias** y haga clic en la flecha para expandir **Protocolos**. Desplácese hacia abajo y seleccione **UDP**. Haga clic en la casilla de verificación **Validate the UDP checksum if possible** (Validar checksum UDP si es posible) y luego en **OK**.




- c. Inicie una captura de Wireshark en la interfaz **H1-eth0**.

- d. Inicie una sesión de tftp de **H2** al servidor tftp en **H1** y obtengan el archivo **my\_tftp\_data**.

```
[root@secOps analyst]# tftp 10.0.0.11 -c get my_tftp_data
```

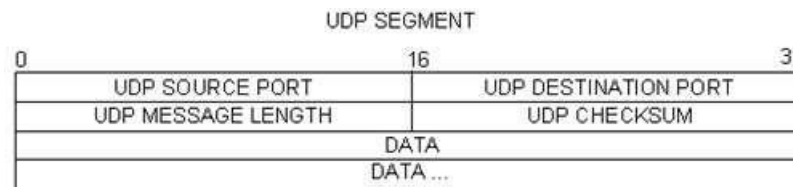
- e. Detenga la captura de Wireshark. Defina el filtro en **tftp** y hagan clic en **Apply** (Aplicar). Utilicen los tres paquetes TFTP para completar la tabla y responder las preguntas del resto de esta práctica de laboratorio.

Filter:	<input type="text" value="tftp"/>		Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.12	10.0.0.11	TFTP	66	Read Request, File: my_tftp_data, Transfer
2	0.001295043	10.0.0.11	10.0.0.12	TFTP	80	Data Packet, Block: 1 (last)
3	0.001735272	10.0.0.12	10.0.0.11	TFTP	46	Acknowledgement, Block: 1

El panel de detalles de paquetes de Wireshark muestra información detallada sobre UDP. Resalte el primer datagrama UDP del equipo host y mueva el puntero del mouse al panel de detalles de paquetes. Puede ser necesario ajustar el panel de detalles del paquete y expandir el registro UDP haciendo clic en la casilla de expansión de protocolo. El datagrama UDP expandido debe ser similar al siguiente diagrama.

UDP Header	User Datagram Protocol, Src Port: 47844, Dst Port: 69 Source Port: 47844 Destination Port: 69 Length: 32 Checksum: 0x2029 [correct] [Checksum Status: Good] [Stream index: 0]
UDP Data	Trivial File Transfer Protocol Opcode: Read Request (1) Source File: my_tftp_data Type: netascii

En la siguiente ilustración, se muestra un diagrama de datagrama UDP. La información del encabezado está dispersa comparada con la del datagrama TCP. Al igual que TCP, cada datagrama UDP se identifica mediante el puerto de origen de UDP y el puerto de destino UDP.



Utilice la captura de Wireshark del primer datagrama UDP para completar la información acerca del encabezado UDP. El valor de checksum es un valor hexadecimal (base 16) indicado por el código anterior 0x:

Descripción	Resultados del Wireshark
Dirección IP de origen	
Dirección IP de destino	
Número de puerto de origen	

Descripción	Resultados del Wireshark
Número de puerto de destino	
Longitud del mensaje UDP	
Checksum de UDP	

¿Cómo verifica UDP la integridad del datagrama?

Examine la primera trama que devuelve el servidor tftpd. Complete la información acerca del encabezado UDP:

Descripción	Resultados del Wireshark
Dirección IP de origen	
Dirección IP de destino	
Número de puerto de origen	
Número de puerto de destino	
Longitud del mensaje UDP	
Checksum de UDP	

Observe que el datagrama UDP devuelto tiene un puerto de origen UDP diferente, pero este puerto de origen se utiliza para el resto de la transferencia TFTP. Dado que no hay una conexión confiable, para mantener la transferencia TFTP, sólo se utiliza el puerto de origen usado para comenzar la sesión TFTP.

También observe que el valor de checksum UDP es incorrecto. Lo más probable es que se deba a la descarga de checksum UDP. Para obtener más información acerca del motivo por el cual sucede esto, realice una búsqueda de “UDP checksum offload”.

### Paso 4: Limpieza

En este paso cerrarán y limpiarán Mininet.

- En el terminal que inició Mininet, introduzcan **quit** en el cursor.

```
mininet> quit
```

- En el prompt, ingrese **sudo mn -c** para limpiar el proceso iniciado por Mininet.

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

### Pregunta de reflexión

Esta práctica de laboratorio brindó la oportunidad de analizar las operaciones de protocolo UDP y TCP de sesiones TFTP y FTP capturadas. ¿En qué se diferencia la manera de administrar la comunicación de TCP con respecto a UDP?