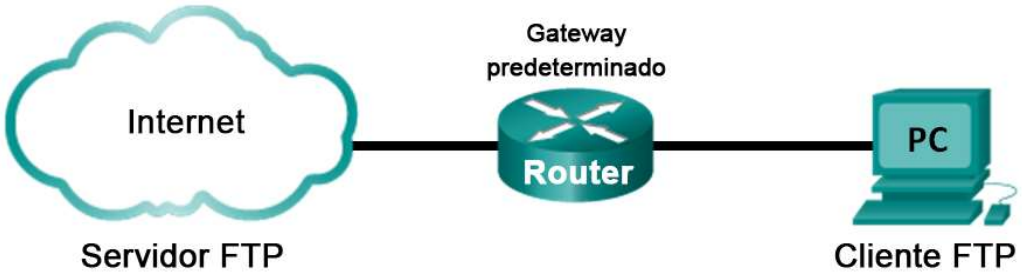
Práctica de laboratorio: Uso de Wireshark para examinar capturas de FTP y TFTP (versión para el instructor)

Nota para el instructor: el color de fuente rojo o las partes resaltadas en gris indican texto que aparece en la copia del instructor solamente.

Topología: parte 1 (FTP)

En la parte 1, se resaltará una captura de TCP de una sesión FTP. Esta topología consta de una PC con acceso a Internet.



Topología: parte 2 (TFTP)

En la parte 2, se resaltará una captura de UDP de una sesión TFTP. La PC debe tener tanto una conexión Ethernet como una conexión de consola para el switch S1.

Tabla de direccionamiento (parte 2)

Dispositivo	Interfaz	Dirección IP	Máscara de subred	Gateway predeterminado
S1	VLAN 1	192.168.1.1	255.255.255.0	No aplicable
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

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

Información básica/Situación

Los dos protocolos en la capa de transporte TCP/IP son TCP, definido en RFC 761, y UDP, definido en RFC 768. Los dos protocolos admiten la comunicación de protocolos de la capa superior. Por ejemplo, TCP se utiliza para proporcionar soporte de la capa de transporte para los protocolos de transferencia de hipertexto (HTTP) y FTP, entre otros. UDP proporciona soporte de la capa de transporte para el Sistema de nombres de dominios (DNS) y TFTP, entre otros.

Nota: entender las partes de los encabezados y de la operación TCP y UDP es una aptitud fundamental con la que deben contar los ingenieros de red.

En la parte 1 de esta práctica de laboratorio, utilizará la herramienta de código abierto de 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. Para conectarse a un servidor FTP anónimo y descargar un archivo, se emplea la utilidad de línea de comandos de Windows. En la parte 2 de esta práctica de laboratorio, utilizará Wireshark para capturar y analizar campos de encabezado del protocolo UDP para las transferencias de archivos TFTP entre el equipo host y el switch S1.

Nota para el instructor: si la versión 1.8.3 o posterior de Wireshark no está cargada en la PC, se puede descargar del URL http://www.wireshark.org/download.html. Para la parte 2 de la práctica de laboratorio, si la versión 4.0 o posterior de tftpd32 no está cargada en la PC, se puede descargar del URL http://tftpd32.jounin.net/tftpd32_download.html.

Nota: el switch que se utiliza es Cisco Catalyst 2960s con Cisco IOS versión 15.0(2) (imagen de lanbasek9). Pueden utilizarse otros switches y versiones de Cisco IOS. Según el modelo y la versión de Cisco IOS, los comandos disponibles y los resultados obtenidos pueden diferir de los que se muestran en las prácticas de laboratorio.

Nota: asegúrese de que el switch se haya borrado y de que no tenga configuraciones de inicio. Si no está seguro, consulte con el instructor.

Nota: en la parte 1, se supone que la PC tiene acceso a Internet, y no se puede realizar utilizando Netlab. La parte 2 es compatible con Netlab.

Nota para el instructor: las instrucciones para borrar el switch se encuentran en el manual de prácticas de laboratorio.

Nota para el instructor: esta práctica de laboratorio se puede realizar en dos sesiones según la disponibilidad de tiempo y equipos. La secuencia de la parte 1 y de la parte 2 no es fundamental.

Nota para el instructor: el uso de un programa detector de paquetes como Wireshark se puede considerar una infracción de la política de seguridad del lugar de estudios. Se recomienda obtener permiso para realizar esta práctica de laboratorio antes de ejecutar Wireshark. Si el uso de un programa detector de paquetes como Wireshark constituye un problema, se sugiere que el instructor asigne la práctica de laboratorio como tarea para el hogar o realice una demostración explicativa.

Recursos necesarios: parte 1 (FTP)

1 PC (Windows 7, Vista o XP con acceso al símbolo del sistema, acceso a Internet y Wireshark instalado)

Recursos necesarios: parte 2 (TFTP)

- 1 switch (Cisco 2960 con Cisco IOS, versión 15.0(2), imagen lanbasek9 o similar)
- 1 PC (Windows 7, Vista o XP con Wireshark y un servidor TFTP, como tftpd32, instalados)
- Cable de consola para configurar los dispositivos Cisco IOS a través del puerto de consola
- Cable Ethernet como se muestra en la topología

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

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

Paso 1: Inicie una captura de Wireshark.

a. Cierre todo el tráfico de la red innecesario, como el explorador Web, para limitar la cantidad de tráfico durante la captura de Wireshark.

b. Inicie la captura de Wireshark.

Paso 2: Descargar el archivo Léame

a. En el símbolo del sistema, introduzca **ftp ftp.cdc.gov**.

b. Conéctese al sitio FTP de Centros para el Control y la Prevención de Enfermedades (CDC) con el usuario **anonymous** y sin contraseña.

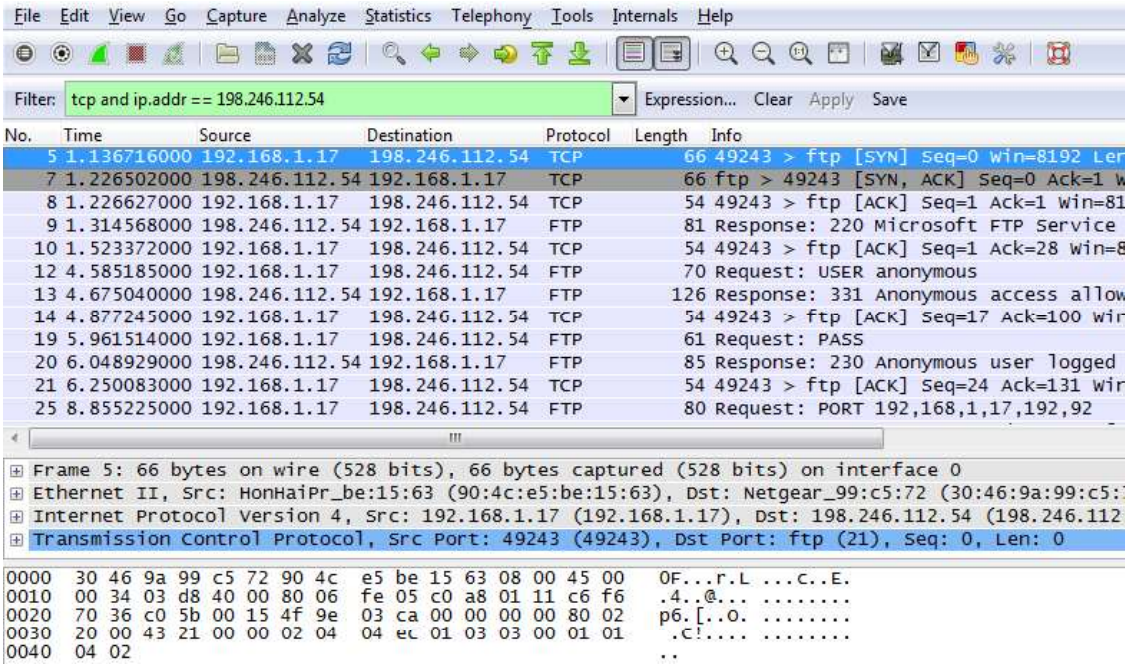
c. Ubique y descargue el archivo Léame.

```
C:\Users\user1>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
aspnet_client
pub
Readme
Siteinfo
up.htm
w3c
web.config
welcome.msg
226 Transfer complete.
ftp: 76 bytes received in 0.00Seconds 19.00Kbytes/sec.
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection for Readme(1428 bytes).
226 Transfer complete.
ftp: 1428 bytes received in 0.01Seconds 204.00Kbytes/sec.
ftp> quit
221
```

Paso 3: Detener la captura de Wireshark

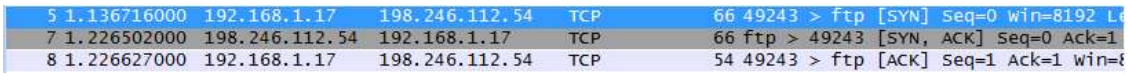
Paso 4: Ver la ventana principal de Wireshark

Wireshark capturó muchos paquetes durante la sesión FTP a ftp.cdc.gov. Para limitar la cantidad de datos para analizar, escriba **tcp and ip.addr == 198.246.112.54** en el área de entrada **Filter:** (Filtrar:) y haga clic en **Apply** (Aplicar). La dirección IP, 198.246.112.54, es la dirección para ftp.cdc.gov.



Paso 5: Analizar los campos TCP

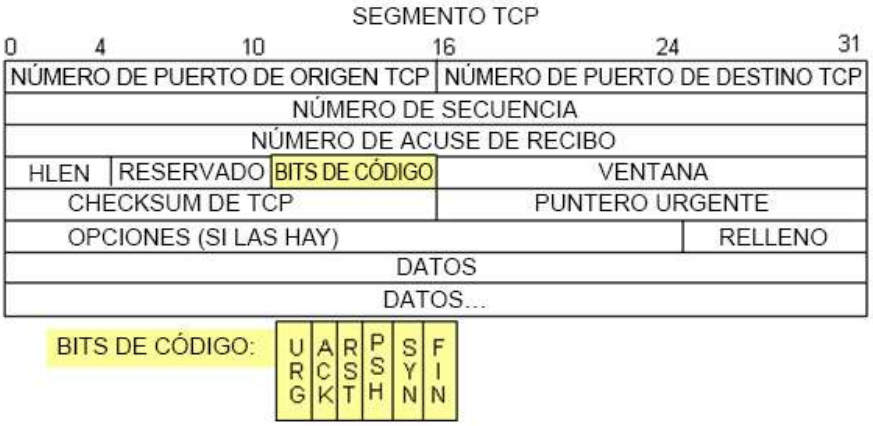
Una vez aplicado el filtro TCP, las primeras tres tramas en el panel de la lista de paquetes (sección superior) muestran el protocolo de la capa de transporte TCP que crea una sesión confiable. La secuencia de [SYN], [SYN, ACK] y [ACK] ilustra el protocolo de enlace de tres vías.



El TCP se utiliza en forma continua durante una sesión para controlar la entrega del datagrama, verificar la llegada del datagrama y administrar el tamaño de la ventana. Por 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. Por último, cuando la sesión FTP finaliza, TCP realiza un cierre y terminación 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 equipo host y expanda el registro TCP. El datagrama TCP expandido parece similar al panel de detalles del paquete que se muestra a continuación.

```
⊞ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54)
⊟ Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 0, Len: 0
    Source port: 49243 (49243)
    Destination port: ftp (21)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊟ 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: 0x4321 [validation disabled]
  ⊞ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), Nc
```



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 inició 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, HTTP, etc.). La combinación de dirección IP de origen, puerto de origen, dirección IP de destino y puerto de destino identifica de manera exclusiva la sesión tanto para el emisor como para el receptor.

Nota: en la captura de Wireshark que se muestra a continuación, el puerto de destino es 21, que es FTP. Los servidores FTP escuchan las conexiones de cliente FTP en el puerto 21.

- El **número de secuencia** especifica el número del último octeto en un segmento.
- El **número de acuse de recibo** especifica el próximo octeto que espera el receptor.
- Los **bits de código** tienen un significado especial en la administración de sesión y en el tratamiento de los segmentos. Entre los valores interesentes se encuentran:

- ACK: acuse de recibo de un segmento.
- SYN: sincronizar; solo está configurado cuando se negocia una sesión TCP nueva durante el protocolo de enlace de tres vías.
- FIN: finalizar; solicitud para cerrar la sesión TCP.

- **Window size** (Tamaño de la ventana) es el valor de la ventana deslizante; determina cuántos octetos se pueden enviar antes de esperar un acuse de recibo.
- **Urgent pointer** (Indicador urgente) se utiliza solo con un indicador urgente (URG) cuando el emisor necesita enviar datos urgentes al receptor.
- En **Options** (Opciones), hay una sola opción actualmente, y se define como el tamaño máximo del segmento TCP (valor optativo).

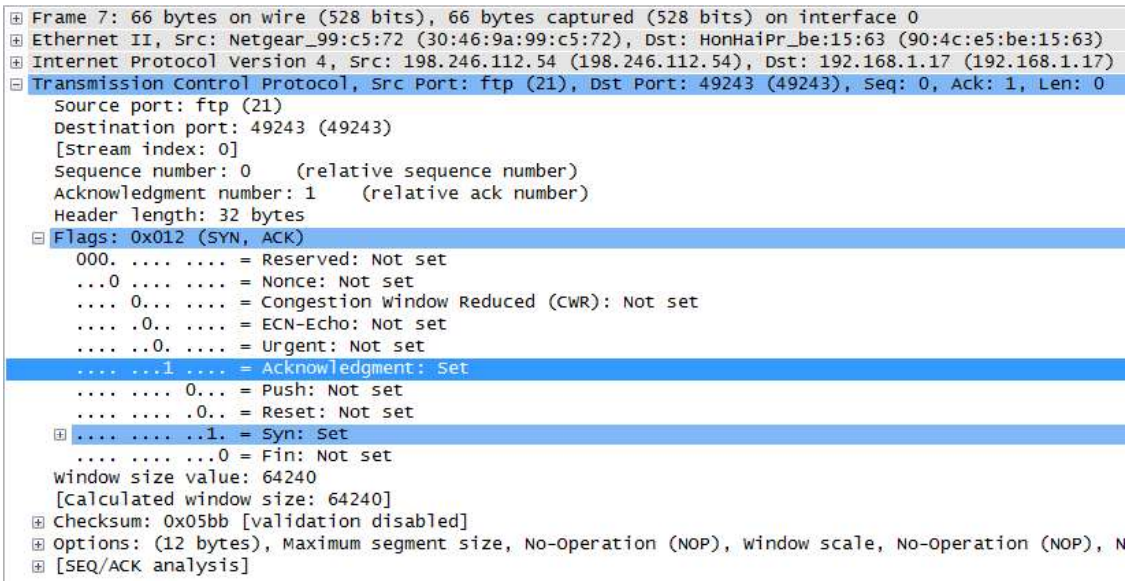
Utilice la captura de Wireshark del inicio de la primera sesión TCP (bit SYN establecido en 1) para completar la información acerca del encabezado TCP:

De la PC al servidor CDC (solo el bit SYN está establecido en 1):

Dirección IP de origen:	192.168.1.17*
Dirección IP de destino:	198.246.112.54
Número de puerto de origen:	49243*
Número de puerto de destino:	21
Número de secuencia:	0 (relativo)
Número de acuse de recibo:	No es aplicable a esta captura
Longitud del encabezado:	32 bytes
Tamaño de la ventana:	8192

*Las respuestas de los estudiantes variarán.

En la segunda captura filtrada de Wireshark, el servidor FTP de CDC acusa recibo de la solicitud de la PC. Observe los valores de los bits SYN y ACK.

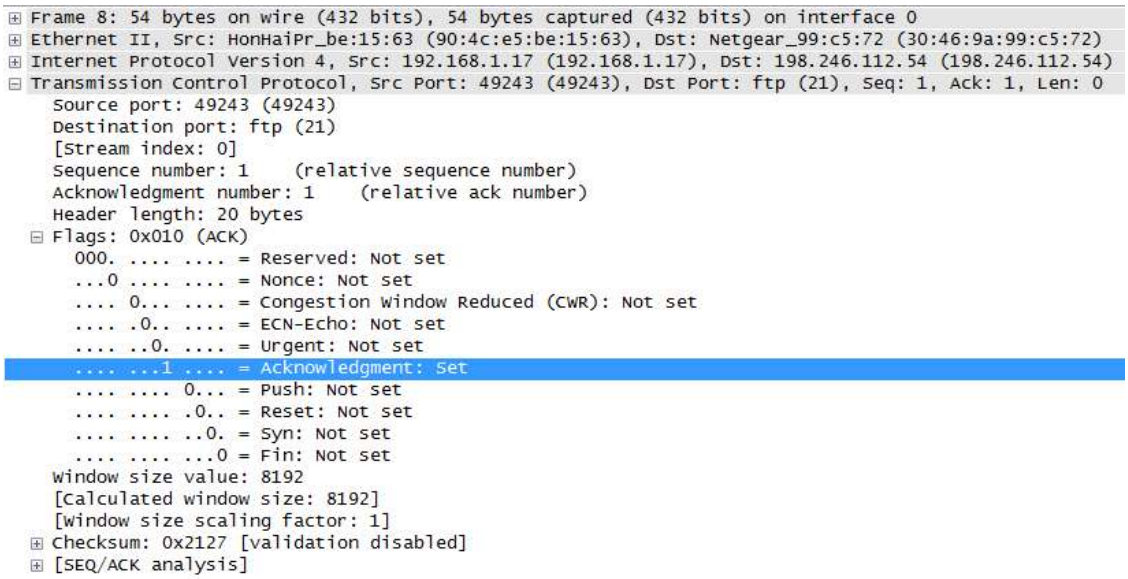


Complete la siguiente información con respecto al mensaje SYN-ACK.

Dirección IP de origen:	198.246.112.54
Dirección IP de destino:	192.168.1.17*
Número de puerto de origen:	21
Número de puerto de destino:	49243*
Número de secuencia:	0 (relativo)
Número de acuse de recibo:	1
Longitud del encabezado:	32 bytes
Tamaño de la ventana:	64240

*Las respuestas de los estudiantes variarán.

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



Complete la siguiente información con respecto al mensaje ACK.

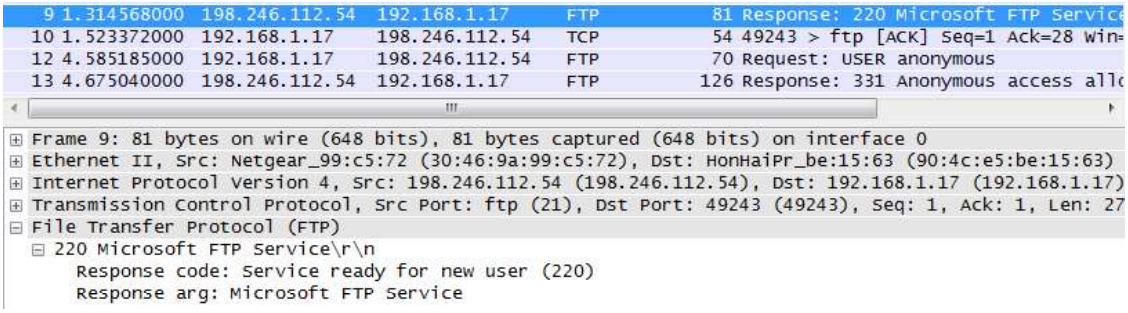
Dirección IP de origen:	192.168.1.17*
Dirección IP de destino:	198.246.112.54
Número de puerto de origen:	49243*
Número de puerto de destino:	21
Número de secuencia:	1
Número de acuse de recibo:	1
Longitud del encabezado:	20
Tamaño de la ventana:	8192*

*Las respuestas de los estudiantes variarán.

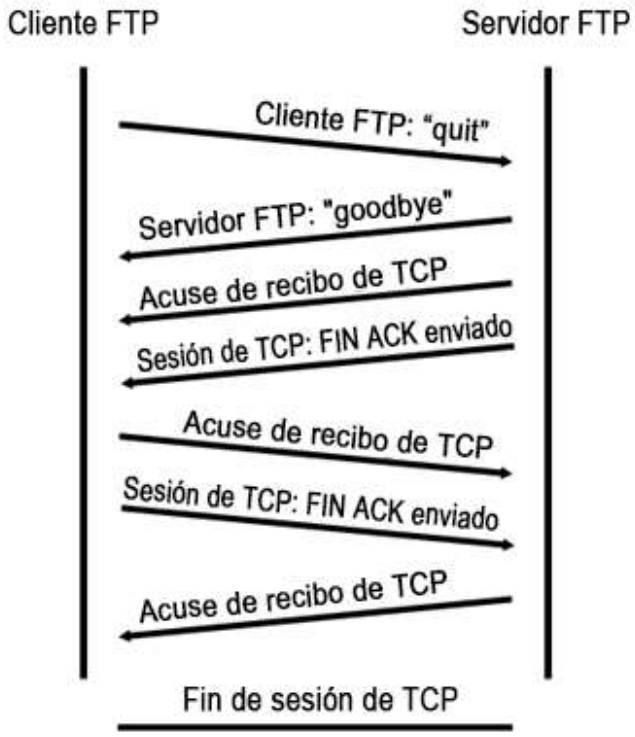
¿Cuántos otros datagramas TCP contenían un bit SYN?

Uno. El primer paquete que envió el host al principio de una sesión TCP.

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 sí sin saber que TCP tiene el control y manejo de la sesión. Cuando el servidor FTP envía una Response: 220 (Respuesta: 220) al cliente FTP, la sesión TCP en el cliente FTP envía un acuse de recibo a la sesión TCP en el servidor. Esta secuencia se puede ver en la captura de Wireshark, a continuación.



Cuando la sesión FTP terminó, el cliente FTP envía un comando para “salir”. El servidor FTP acusa recibo de la terminación FTP con una Response: 221 Goodbye (Respuesta: 221. Adiós). En este momento, la sesión TCP del servidor FTP envía un datagrama TCP al cliente FTP, en el que se anuncia la terminación de la sesión TCP. La sesión TCP del cliente FTP acusa recibo de 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 acusar recibo de la terminación y se cierra la sesión TCP. Esta secuencia se puede ver en el diagrama y la captura que se muestran a continuación.



Si se aplica un filtro **ftp**, puede examinarse la secuencia completa del tráfico FTP en Wireshark. Observe la secuencia de eventos durante esta sesión FTP. Para recuperar el archivo Léame, se utilizó el nombre de usuario anónimo. Una vez que se completó la transferencia de archivos, el usuario finalizó la sesión FTP.

Filter: ftp — Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
9	1.314568000	198.246.112.54	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
12	4.585185000	192.168.1.17	198.246.112.54	FTP	70	Request: USER anonymous
13	4.675040000	198.246.112.54	192.168.1.17	FTP	126	Response: 331 Anonymous access allowe
19	5.961514000	192.168.1.17	198.246.112.54	FTP	61	Request: PASS
20	6.048929000	198.246.112.54	192.168.1.17	FTP	85	Response: 230 Anonymous user logged i
25	8.855225000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,92
26	8.945530000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
27	8.955549000	192.168.1.17	198.246.112.54	FTP	60	Request: NLST
29	9.053034000	198.246.112.54	192.168.1.17	FTP	109	Response: 150 Opening ASCII mode data
39	9.347432000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
42	12.621720000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,93
43	12.709658000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
44	12.722592000	192.168.1.17	198.246.112.54	FTP	67	Request: RETR Readme
45	12.811097000	198.246.112.54	192.168.1.17	FTP	118	Response: 150 Opening ASCII mode data
58	13.107294000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
61	15.514815000	192.168.1.17	198.246.112.54	FTP	60	Request: QUIT
62	15.601920000	198.246.112.54	192.168.1.17	FTP	61	Response: 221

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. Dado que la conexión TCP es full-duplex, cada dirección debe terminar independientemente. Examine las direcciones de origen y destino.

En este ejemplo, el servidor FTP no tiene más datos para enviar en el stream; envía un segmento con el conjunto de indicadores FIN en la trama 63. La PC envía un ACK para acusar recibo del FIN para terminar la sesión del servidor al cliente en la trama 64.

En la trama 65, la PC envía un FIN al servidor FTP para terminar la sesión TCP. El servidor FTP responde con un ACK para acusar recibo del FIN de la PC en la trama 67. Ahora, la sesión TCP terminó entre el servidor FTP y la PC.

Práctica de laboratorio: Uso de Wireshark para examinar capturas de FTP y TFTP