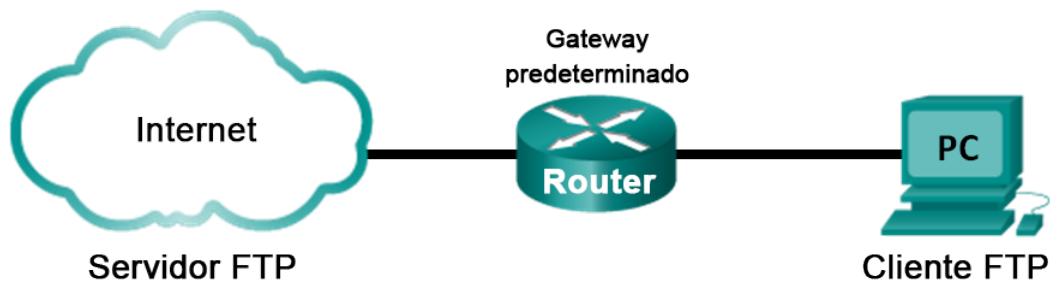


Práctica de laboratorio: Uso de Wireshark para examinar capturas de FTP y TFTP

Topología: parte 1 (FTP)

En la parte 1, se resaltarán 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 resaltarán 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: 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.

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.

The screenshot shows the Wireshark interface with the following details:

- Filter:** tcp and ip.addr == 198.246.112.54
- Packet List:**

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|----------------|----------------|----------|--------|---|
| 5 | 1.136716000 | 192.168.1.17 | 198.246.112.54 | TCP | 66 | 49243 > ftp [SYN] Seq=0 win=8192 Len=0 |
| 7 | 1.226502000 | 198.246.112.54 | 192.168.1.17 | TCP | 66 | ftp > 49243 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 |
| 8 | 1.226627000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 9 | 1.314568000 | 198.246.112.54 | 192.168.1.17 | FTP | 81 | Request: 220 Microsoft FTP Service |
| 10 | 1.523372000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=1 Ack=28 Win=8192 Len=0 |
| 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 allowed |
| 14 | 4.877245000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=17 Ack=100 Win=8192 Len=0 |
| 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 in |
| 21 | 6.250083000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=24 Ack=131 Win=8192 Len=0 |
| 25 | 8.855225000 | 192.168.1.17 | 198.246.112.54 | FTP | 80 | Request: PORT 192,168,1,17,192,92 |
- Packet Details:**
 - 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
- Packet Bytes:**

```
0000  30 46 9a 99 c5 72 90 4c  e5 be 15 63 08 00 45 00  0F...r.L ...C..E.
0010  00 34 03 d8 40 00 80 06  fe 05 c0 a8 01 11 c6 f6  .4...@.....
0020  70 36 c0 5b 00 15 4f 9e  03 ca 00 00 00 00 80 02  p6.[...O.....
0030  20 00 43 21 00 00 02 04  04 ec 01 03 03 00 01 01  .C!.....
0040  04 02
```

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.

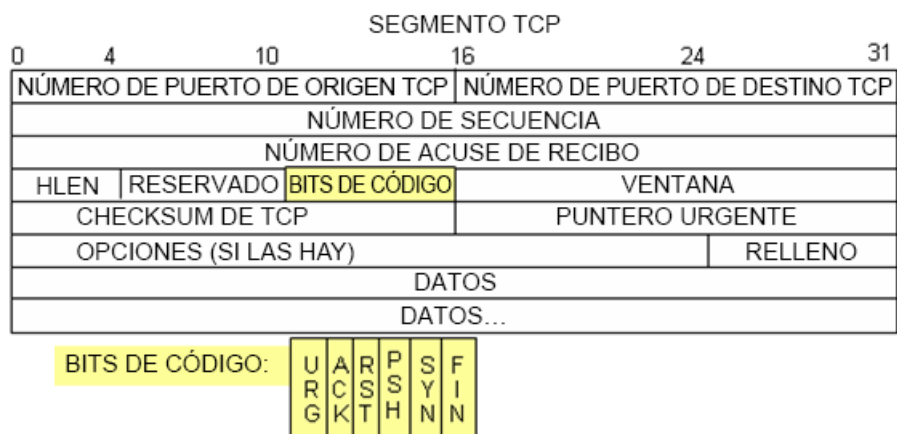
| | | | | | | |
|---|-------------|----------------|----------------|-----|----|---|
| 5 | 1.136716000 | 192.168.1.17 | 198.246.112.54 | TCP | 66 | 49243 > ftp [SYN] Seq=0 win=8192 Len=0 |
| 7 | 1.226502000 | 198.246.112.54 | 192.168.1.17 | TCP | 66 | ftp > 49243 [SYN, ACK] Seq=0 Ack=1 Len=0 |
| 8 | 1.226627000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=1 Ack=1 win=0 Len=0 |

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
    ....0... = 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), No
  
```



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 interesantes 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: | |
| 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: | |

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.

```

+ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
+ Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63)
+ Internet Protocol Version 4, Src: 198.246.112.54 (198.246.112.54), Dst: 192.168.1.17 (192.168.1.17)
- Transmission Control Protocol, Src Port: ftp (21), Dst Port: 49243 (49243), Seq: 0, Ack: 1, Len: 0
  Source port: ftp (21)
  Destination port: 49243 (49243)
  [Stream index: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header length: 32 bytes
  - 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: 64240
  [Calculated window size: 64240]
  + Checksum: 0x05bb [validation disabled]
  + Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), N
  + [SEQ/ACK analysis]

```

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

| | |
|------------------------------|--|
| 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: | |

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.

```

+ Frame 8: 54 bytes on wire (432 bits), 54 bytes captured (432 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: 1, Ack: 1, Len: 0
  Source port: 49243 (49243)
  Destination port: ftp (21)
  [Stream index: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header length: 20 bytes
  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: 0x2127 [validation disabled]
  [SEQ/ACK analysis]
```

Complete la siguiente información con respecto al mensaje ACK.

| | |
|------------------------------|--|
| 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: | |

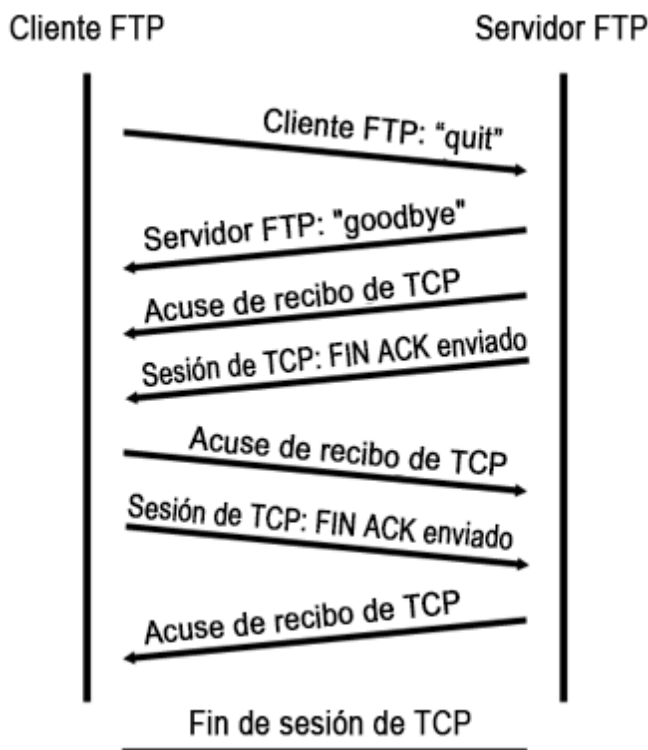
¿Cuántos otros datagramas TCP contenían un bit SYN?

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 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.

| | | | | | | |
|----|-------------|----------------|----------------|-----|-----|---------------------------------------|
| 9 | 1.314568000 | 198.246.112.54 | 192.168.1.17 | FTP | 81 | Response: 220 Microsoft FTP Service |
| 10 | 1.523372000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=1 Ack=28 win= |
| 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 |

| | |
|---|--|
| Frame 9: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0 | |
| Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63) | |
| Internet Protocol Version 4, Src: 198.246.112.54 (198.246.112.54), Dst: 192.168.1.17 (192.168.1.17) | |
| Transmission Control Protocol, Src Port: ftp (21), Dst Port: 49243 (49243), 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 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.

| 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 allowed |
| 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 in |
| 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.

| | | | | | | |
|----|--------------|----------------|----------------|-----|----|-------------------------------------|
| 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 |
| 63 | 15.602245000 | 198.246.112.54 | 192.168.1.17 | TCP | 54 | ftp > 49243 [FIN, ACK] Seq=365 Ack= |
| 64 | 15.602314000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [ACK] Seq=101 Ack=366 v |
| 65 | 15.605832000 | 192.168.1.17 | 198.246.112.54 | TCP | 54 | 49243 > ftp [FIN, ACK] Seq=101 Ack= |
| 67 | 15.696497000 | 198.246.112.54 | 192.168.1.17 | TCP | 54 | ftp > 49243 [ACK] Seq=366 Ack=102 v |

Frame 63: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
 Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63)
 Internet Protocol Version 4, Src: 198.246.112.54 (198.246.112.54), Dst: 192.168.1.17 (192.168.1.17)
 Transmission Control Protocol, Src Port: ftp (21), Dst Port: 49243 (49243), Seq: 365, Ack: 101, Len

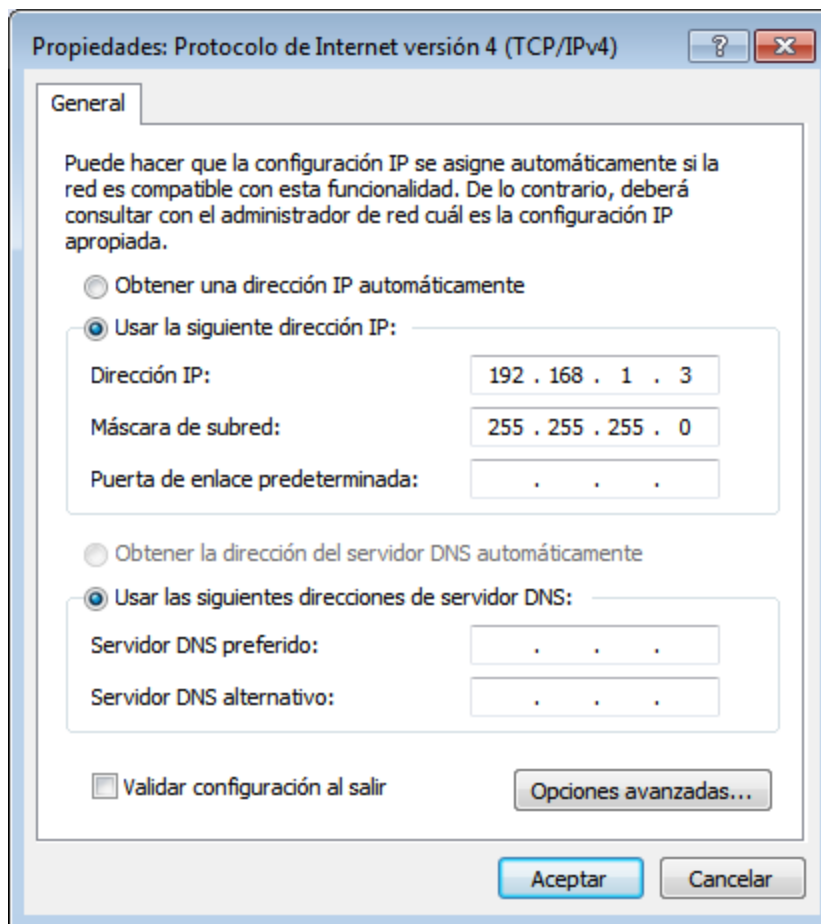
Parte 2: Identificar campos de encabezado y operación UDP mediante una captura de sesión TFTP de Wireshark

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

Paso 1: Configurar esta topología física y prepararse para la captura de TFTP



- Establezca una consola y una conexión Ethernet entre la PC-A y el switch S1.
- Si aún no lo hizo, configure manualmente la dirección IP de la PC en 192.168.1.3. No se requiere configurar el gateway predeterminado.



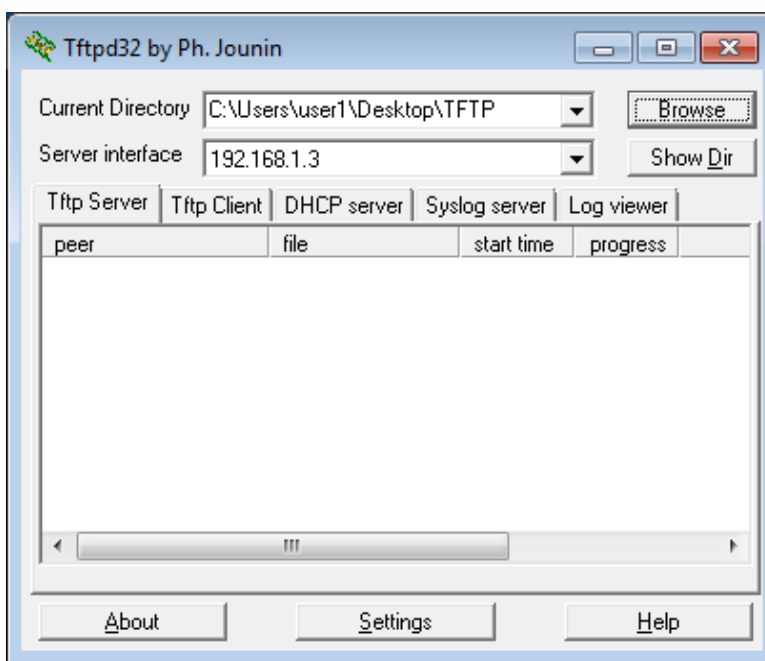
- c. Configure el switch. Asigne la dirección IP 192.168.1.1 a VLAN 1. Verifique la conectividad con la PC haciendo ping a 192.168.1.3. Resuelva cualquier problema que se presente.

```
Switch> enable
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.1 255.255.255.0
S1(config-if)# no shut
*Mar 1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar 1 00:37:50.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed state to up
S1(config-if)# end
S1# ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
```

Paso 2: Preparar el servidor TFTP en la PC

- Si aún no existe, cree una carpeta en el escritorio de la PC con el nombre **TFTP**. Los archivos del switch se copiarán a esta ubicación.
- Inicie **tftpd32** en la PC.
- Haga clic en **Browse** (Buscar), cambie el directorio actual por **C:\Usuarios\usuario1\Escritorio\TFTP** y reemplace usuario1 con su nombre de usuario.

El servidor TFTP debería verse así:



Observe que, en Current Directory (Directorio actual), se indica la interfaz de usuario y servidor (PC-A) como la dirección IP **192.168.1.3**.

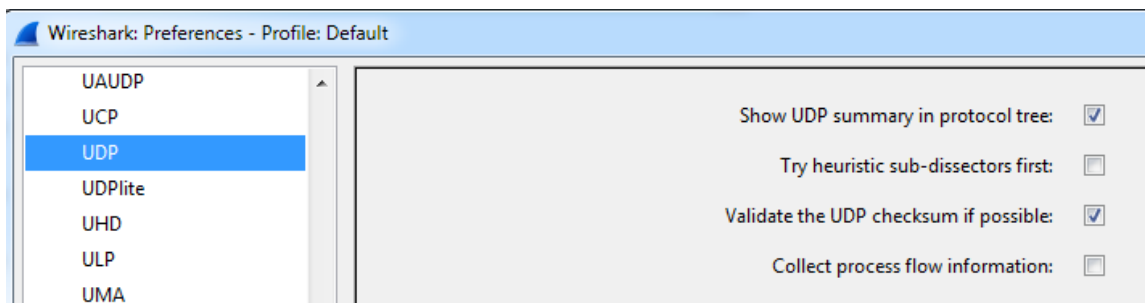
- Pruebe la capacidad de copiar un archivo del switch a la PC con TFTP. Resuelva cualquier problema que se presente.

```
S1# copy start tftp
Address or name of remote host []? 192.168.1.3
Destination filename [s1-config]?
!!
1638 bytes copied in 0.026 secs (63000 bytes/sec)
```

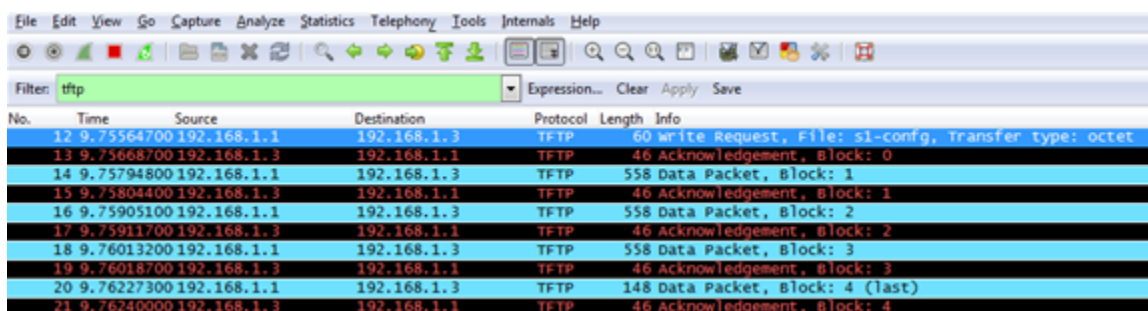
Si ve que el archivo se copió (como en el resultado de arriba), entonces está listo para avanzar al siguiente paso. De lo contrario, resuelva el problema. Si recibe el mensaje de error `%Error opening tftp (Permission denied)` (Error al abrir tftp [permiso denegado]), primero asegúrese de que el firewall no esté bloqueando el TFTP y de que esté copiando a una ubicación donde su nombre de usuario tiene el permiso adecuado, como el escritorio.

Paso 3: Capturar una sesión TFTP en Wireshark

- Abra Wireshark. En el menú **Edit** (Editar), seleccione **Preferences** (Preferencias) y haga clic en el signo más (+) para expandir **Protocols** (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 **Apply** (Aplicar). A continuación, haga clic en **OK** (Aceptar).



- b. Inicie una captura de Wireshark.
- c. Ejecute el comando `copy start tftp` en el switch.
- d. Detener la captura de Wireshark

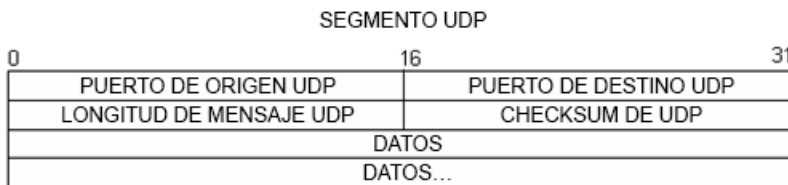


- e. Defina el filtro en **tftp**. El resultado debe ser similar al que se muestra más arriba. Esta transferencia TFTP se utiliza para analizar el funcionamiento de la capa de transporte UDP.

En Wireshark, se encuentra disponible información detallada sobre UDP en el panel de detalles del paquete. Resalte el primer datagrama UDP del equipo host y mueva el puntero del mouse hacia el panel de detalles del paquete. Puede ser necesario ajustar el panel de detalles del paquete y expandir el registro UDP con un clic en la casilla de expansión de protocolo. El datagrama UDP expandido debe ser similar al diagrama siguiente.

| | |
|-------------------|--|
| Encabezado UDP | <div> <div></div> <div>User Datagram Protocol, Src Port: 62513 (62513), Dst Port: tftp (69)</div> </div> |
| | <div> <div></div> <div>Source port: 62513 (62513)</div> <div></div> <div>Destination port: tftp (69)</div> <div></div> <div>Length: 25</div> <div></div> <div>Checksum: 0x482c [correct]</div> </div> |
| UDP Data | <div> <div></div> <div>Trivial File Transfer Protocol</div> <div></div> <div>[DESTINATION File: s1-config]</div> <div></div> <div>Opcode: Write Request (2)</div> <div></div> <div>DESTINATION File: s1-config</div> <div></div> <div>Type: octet</div> </div> |

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 está identificado con el puerto de origen UDP y el puerto de destino UDP.



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

| | |
|------------------------------|--|
| Dirección IP de origen: | |
| Dirección IP de destino: | |
| Número de puerto de origen: | |
| Número de puerto de destino: | |
| Longitud de mensaje UDP: | |
| Checksum de UDP: | |

¿Cómo verifica UDP la integridad del datagrama?

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

| | |
|------------------------------|--|
| Dirección IP de origen: | |
| Dirección IP de destino: | |
| Número de puerto de origen: | |
| Número de puerto de destino: | |
| Longitud de mensaje UDP: | |
| Checksum de UDP: | |

- ▣ User Datagram Protocol, Src Port: 58565 (58565), Dst Port: 62513 (62513)
 - Source port: 58565 (58565)
 - Destination port: 62513 (62513)
 - Length: 12
 - ▣ Checksum: 0x8372 [incorrect, should be 0xa385 (maybe caused by "UDP checksum offload"?)]
- ▣ Trivial File Transfer Protocol
 - [DESTINATION File: s1-config]
 - Opcode: Acknowledgement (4)
 - Block: 0

Observe que el datagrama UDP devuelto tiene un puerto de origen UDP diferente, pero este puerto de origen es utilizado para el resto de la transferencia TFTP. Dado que no hay una conexión confiable, para mantener la transferencia TFTP, se utiliza solo el puerto de origen que se empleó para iniciar 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".

Reflexión

Esta práctica de laboratorio brindó a los estudiantes la oportunidad de analizar las operaciones de los protocolos UDP y TCP de las sesiones TFTP y FTP capturadas. ¿De qué manera TCP administra la comunicación distinto de como lo hace UDP?

Desafío

Debido a que ni FTP ni TFTP son protocolos seguros, todos los datos transferidos se envían en texto no cifrado. Esto incluye cualquier ID de usuario, contraseñas o contenido de archivo en texto no cifrado. Si analiza la sesión FTP de capa superior, identificará rápidamente la ID de usuario, la contraseña y las contraseñas de archivos de configuración. El examen de datos TFTP de capa superior es un poco más complicado, pero se puede examinar el campo de datos y extraer información sobre la ID de usuario de configuración y la contraseña.

Limpieza

Salvo que el instructor indique lo contrario:

- 1) Elimine los archivos que se copiaron a su PC.
- 2) Borre las configuraciones del switch **S1**.
- 3) Elimine la dirección IP manual de la PC y restaure la conectividad a Internet.