

JULIO ANTHONY ENGELS RUIZ COTO – 1284719

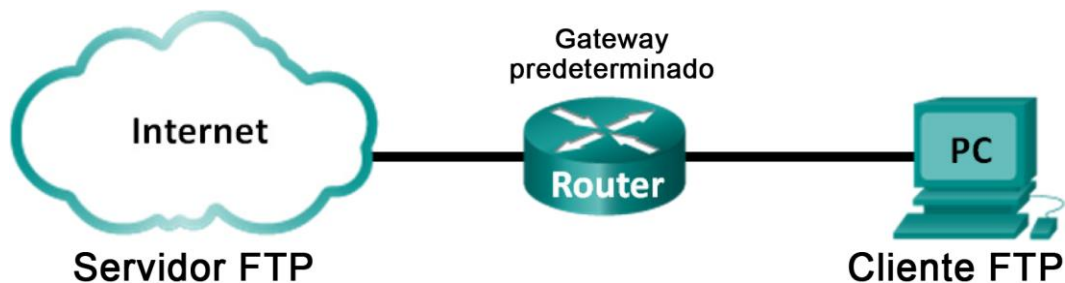
EDDIE ALEJANDRO GIRON CARRAZA - 1307419

Laboratorio No. 9 – Parte 1

Práctica de laboratorio: Uso de 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 consta de una PC con acceso a Internet.



Topología: Parte 2 (TFTP)

La parte 2 destacará una captura de UDP de una sesión TFTP. La PC debe tener una conexión Ethernet y una conexión de consola al switch S1.



Tabla de direccionamiento (parte 2)

Dispositivo	Interfaces	Dirección IP	Máscara de subred	Gateway predeterminado
S1	VLAN 1	192.168.1.1	255.255.255.0	N/D
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.

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.

Nota: Comprender las partes de los encabezados y de la operación de TCP y UDP es una habilidad fundamental para los ingenieros de red.

En la parte 1 de esta práctica de laboratorio, utilizará 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. 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 UDP para las transferencias de archivos TFTP entre el equipo host y S1.

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

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

Recursos necesarios: Parte 1 (FTP)

1 PC (Windows 7 u 8 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 comparable)
- 1 PC (Windows 7 u 8 con Wireshark y un servidor TFTP, como tftpd32, instalados)
- Cable de consola para configurar los dispositivos con Cisco IOS mediante los puertos 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, utilizará Wireshark para capturar una sesión FTP e inspeccionar los campos de encabezado de TCP.

Paso 1: Iniciar una captura de Wireshark.

- a. Cierre todo el tráfico de red innecesario, como el navegador 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.dlptest.com**.
- b. Conéctese al sitio FTP "DLP TEST" con el usuario **dlpuser** y contraseña **rNrKYTX9g7z3RgJRMxWuGHbeu**.

```
PS C:\Users\djdonis> ftp ftp.dlptest.com
Conectado a ftp.dlptest.com.
220 Welcome to the DLP Test FTP Server
200 Always in UTF8 mode.
Usuario (ftp.dlptest.com:(none)): dlpuser
331 Please specify the password.
Contraseña:
230 Login successful.
ftp>
```

- c. Localice y descargue el archivo “NetAllyTest.txt” usando el comando **ls** para mostrar los archivos.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
10
NetAllyTest.txt
Vorne
input
226 Directory send OK.
ftp: 38 bytes recibidos en 0.02segundos 2.38a KB/s.
ftp>
```

- d. Introduzca el comando **get NetAllyTest.txt** para descargar el archivo. Cuando finalice la descarga, introduzca el comando **quit** para salir.

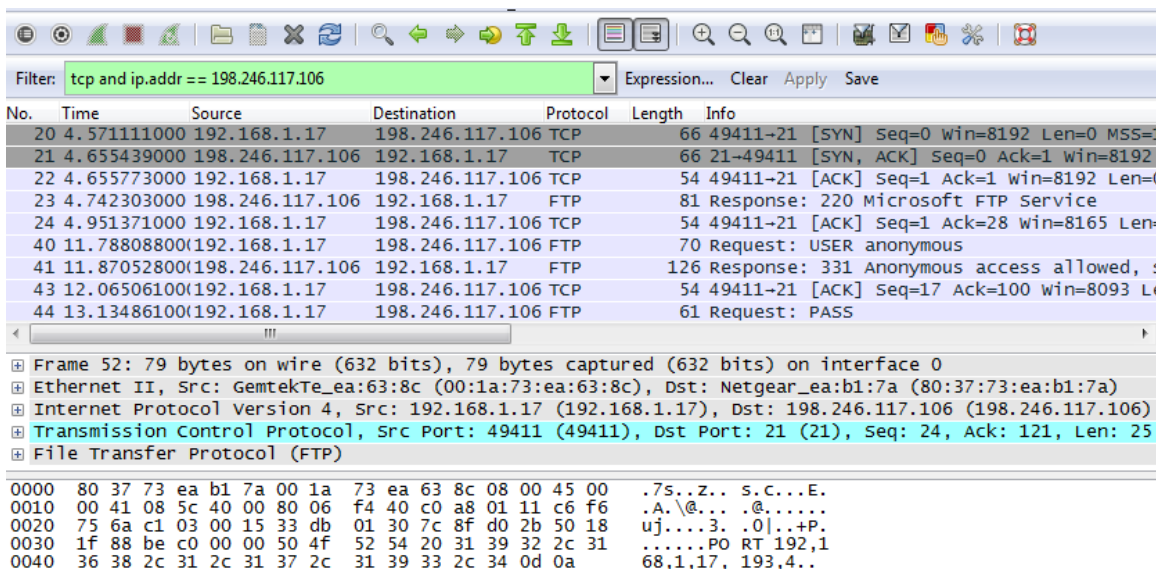
```
ftp> get NetAllyTest.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for NetAllyTest.txt (10485760 bytes).
226 Transfer complete.
ftp: 10485760 bytes recibidos en 12.74segundos 822.93a KB/s.
ftp> quit
221 Goodbye.
```

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. Para limitar la cantidad de datos para el análisis, escriba **tcp and ip.addr == 44.241.66.173** en el área de entrada **Filter:** (Filtrar) y haga clic en **Apply** (Aplicar). La dirección IP, **44.241.66.173**, es la dirección para ftp.dlptest.com en este momento.

Práctica de laboratorio: Uso de Wireshark para examinar capturas de TCP y UDP



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.

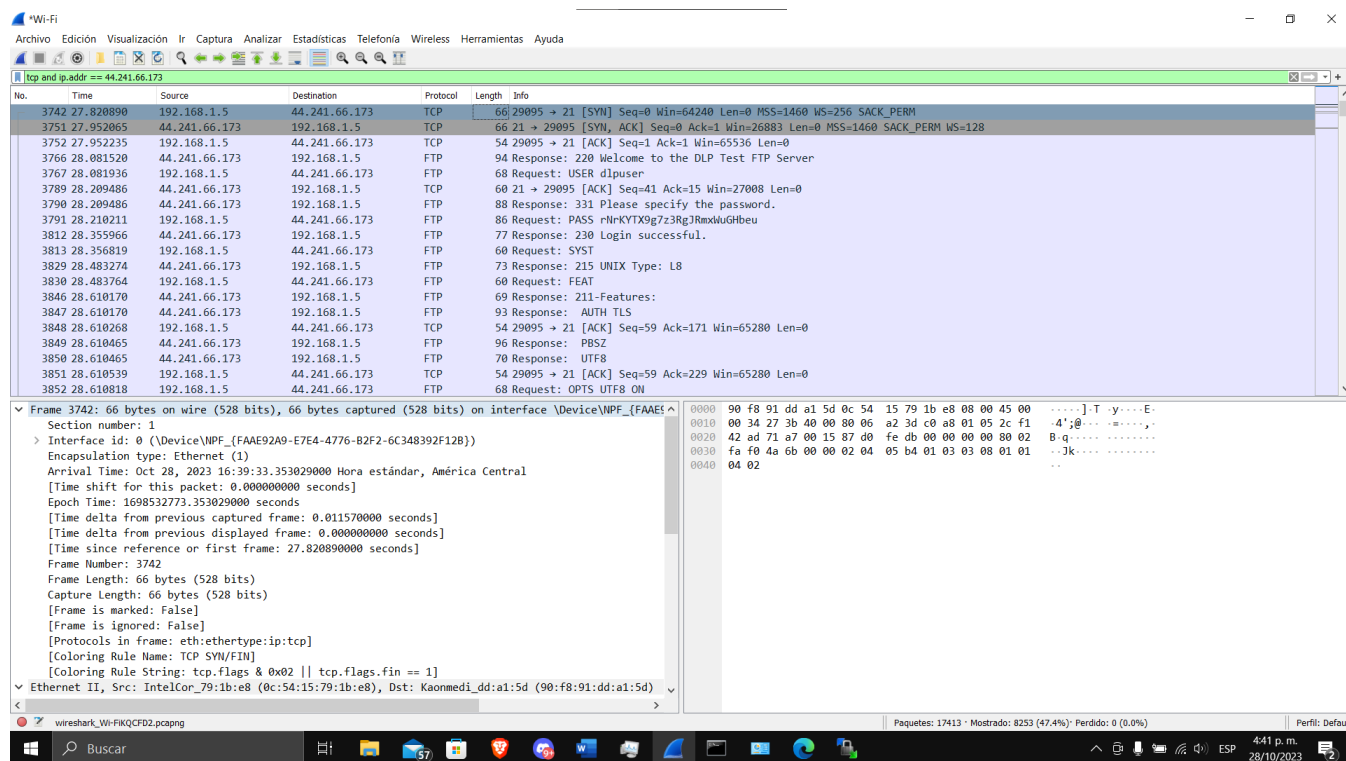
0010 00 41 08 5c 40 00 80 06 f4 40 c0 a8 01 11 c6 f6 .A.\@... .@.....

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 31PO RT 192,1

0040 36 38 2c 31 2c 31 37 2c 31 39 33 2c 34 0d 0a 68,1,17, 193,4..

Colocar captura de pantalla



*Wi-Fi

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

tcp and ip.addr == 44.241.66.173

No.	Time	Source	Destination	Protocol	Length	Info
3742	27.820890	192.168.1.5	44.241.66.173	TCP	66	29095 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3751	27.952065	44.241.66.173	192.168.1.5	TCP	66	21 → 29095 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM WS=128
3752	27.952235	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0
3766	28.081520	44.241.66.173	192.168.1.5	FTP	94	Response: 220 Welcome to the DLP Test FTP Server
3767	28.081936	192.168.1.5	44.241.66.173	FTP	68	Request: USER dlpuser
3789	28.209486	44.241.66.173	192.168.1.5	TCP	60	21 → 29095 [ACK] Seq=41 Ack=15 Win=27008 Len=0
3790	28.209486	44.241.66.173	192.168.1.5	FTP	88	Response: 331 Please specify the password.
3791	28.210211	192.168.1.5	44.241.66.173	FTP	86	Request: PASS rNrKXTX9g7z3RgJRMxluGHbeu
3812	28.359966	44.241.66.173	192.168.1.5	FTP	77	Response: 230 Login successful.
3813	28.356819	192.168.1.5	44.241.66.173	FTP	60	Request: SYST
3829	28.483274	44.241.66.173	192.168.1.5	FTP	73	Response: 215 UNIX Type: L8
3830	28.483764	192.168.1.5	44.241.66.173	FTP	60	Request: FEAT
3846	28.618170	44.241.66.173	192.168.1.5	FTP	69	Response: 211-Features:
3847	28.618170	44.241.66.173	192.168.1.5	FTP	93	Response: AUTH TLS
3848	28.618268	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=59 Ack=171 Win=65280 Len=0
3849	28.618465	44.241.66.173	192.168.1.5	FTP	96	Response: PBSZ
3850	28.618465	44.241.66.173	192.168.1.5	FTP	70	Response: UTF8
3851	28.618539	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=59 Ack=229 Win=65280 Len=0
3852	28.618818	192.168.1.5	44.241.66.173	FTP	68	Request: OPTS UTF8 ON

Frame 3742: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{FAAE92A9-E7E4-4776-B2F2-6C348392F128}

Section number: 1

Interface id: 0 (\Device\NPF_{FAAE92A9-E7E4-4776-B2F2-6C348392F128})

Encapsulation type: Ethernet (1)

Arrival Time: Oct 28, 2023 16:39:33.353029000 Hora estándar, América Central

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1698532773.353029000 seconds

[Time delta from previous captured frame: 0.011570000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 27.820890000 seconds]

Frame Number: 3742

Frame Length: 66 bytes (528 bits)

Capture Length: 66 bytes (528 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: ethertypertype:ip:tcp]

[Coloring Rule Name: TCP SYN/FIN]

[Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]

Ethernet II, Src: IntelCor_79:1b:e8 (0c:54:15:79:1b:e8), Dst: Kaonmedi_dd:a1:5d (90:f8:91:dd:a1:5d)

0000 90 f8 91 dd a1 5d 0c 54 15 79 1b e8 08 00 45 00T.y....E.

0010 00 34 27 3b 40 00 00 06 a2 3d c0 a8 01 05 2c f1 4's@...

0020 42 ad 71 a7 00 15 87 d0 fe db 00 00 00 00 02 B-q.....

0030 fa f0 4a 6b 00 00 02 04 05 b4 01 03 03 08 01 01 ..Jk.....

0040 04 02 ..

Paquetes: 17413 - Mostrado: 8253 (47.4%) - Perdido: 0 (0.0%) Perfil: Default

441 p.m. 28/10/2023

Paso 5: Analizar los campos TCP.

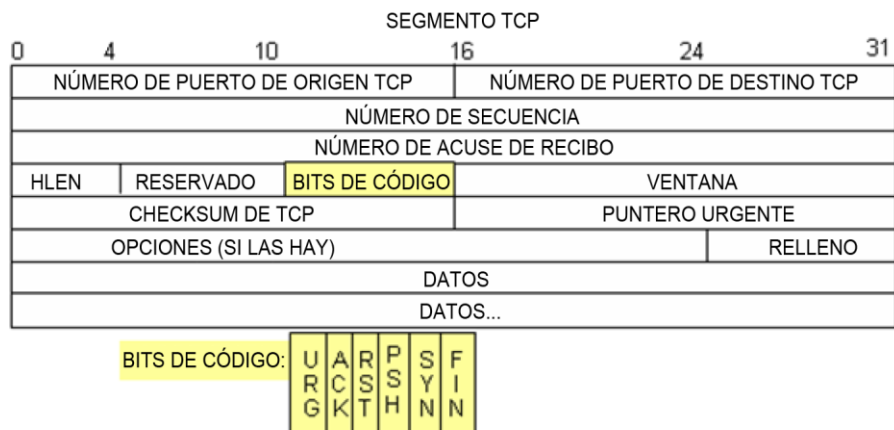
Una vez aplicado el filtro TCP, las primeras tres tramas en el panel de la lista de paquetes (sección superior) muestran el protocolo de la capa de transporte TCP que crea una sesión confiable. La secuencia de [SYN], [SYN, ACK] y [ACK] ilustra el protocolo de enlace de tres vías.

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 el primer datagrama TCP del equipo host y expanda el datagrama TCP. El datagrama TCP expandido se muestra de manera similar al panel de detalles de paquetes que 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-Operation (NOP)	



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 o la aplicación de capa superior 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 siguiente captura de Wireshark, el puerto de destino es 21, que es 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 configura cuando se negocia una nueva sesión TCP durante el protocolo de enlace de tres vías TCP.
 - FIN: finalizar, la solicitud para cerrar la sesión 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.

De la PC al servidor DLPTEST (solo el bit de SYN está configurado en 1):

Dirección IP de origen	192.168.1.1
Dirección IP de destino	44.241.66.173
Número de puerto de origen	29095
Número de puerto de destino	21
Número de secuencia	0 relative sequence number
Número de reconocimiento	0 row
Longitud del encabezado	32 bytes , 8 * 4
Tamaño de la ventana	64240

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

Práctica de laboratorio: Uso de Wireshark para examinar capturas de TCP y UDP

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]

Colocar captura de pantalla

CapturaParte1.pcapng

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

tcp and ip.addr == 44.241.66.173

No.	Time	Source	Destination	Protocol	Length	Info
3742	27.820890	192.168.1.5	44.241.66.173	TCP	66	29095 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3751	27.952065	44.241.66.173	192.168.1.5	TCP	66	21 → 29095 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM WS=128
3752	27.952235	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0
3766	28.081520	44.241.66.173	192.168.1.5	FTP	94	Response: 220 Welcome to the DLP Test FTP Server
3767	28.081936	192.168.1.5	44.241.66.173	FTP	68	Request: USER dlpuser
3789	28.209486	44.241.66.173	192.168.1.5	TCP	60	21 → 29095 [ACK] Seq=41 Ack=15 Win=27008 Len=0
3790	28.209486	44.241.66.173	192.168.1.5	FTP	88	Response: 331 Please specify the password.
3791	28.210211	192.168.1.5	44.241.66.173	FTP	86	Request: PASS rHrKYTX9g7z3RgJrmXluGHbeu
3812	28.355966	44.241.66.173	192.168.1.5	FTP	77	Response: 230 Login successful.
3813	28.356819	192.168.1.5	44.241.66.173	FTP	60	Request: SYST
3829	28.483274	44.241.66.173	192.168.1.5	FTP	73	Response: 215 UNIX Type: L8
3830	28.483764	192.168.1.5	44.241.66.173	FTP	60	Request: FEAT
3846	28.610170	44.241.66.173	192.168.1.5	FTP	69	Response: 211-Features:
3847	28.610170	44.241.66.173	192.168.1.5	FTP	93	Response: AUTH TLS
3848	28.610268	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=59 Ack=171 Win=65280 Len=0
3849	28.610465	44.241.66.173	192.168.1.5	FTP	96	Response: PBSZ
3850	28.610465	44.241.66.173	192.168.1.5	FTP	70	Response: UTF8
3851	28.610539	192.168.1.5	44.241.66.173	TCP	54	29095 → 21 [ACK] Seq=59 Ack=229 Win=65280 Len=0
3852	28.610818	192.168.1.5	44.241.66.173	FTP	68	Request: OPTS UTF8 ON

Frame 3742: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF... [FAAE...]

Ethernet II, Src: IntelCor_79:1b:e8 (0c:54:15:79:1b:e8), Dst: Kaonmedi_dd:a1:5d (90:f8:91:dd:a1:5d)

Internet Protocol Version 4, Src: 192.168.1.5, Dst: 44.241.66.173

Transmission Control Protocol, Src Port: 29095, Dst Port: 21, Seq: 0, Len: 0

Source Port: 29095

Destination Port: 21

[Stream index: 18]

[Conversation completeness: Incomplete, DATA (15)]

[TCP segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 2278620891

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 0

Acknowledgment number (raw): 0

1000 = Header Length: 32 bytes (8)

Flags: 0x002 (SYN)

Window: 64240

[Calculated window size: 64240]

Checksum: 0x4a6b [unverified]

Paquetes: 17413 · Mostrado: 8253 (47.4%) · Perdido: 0 (0.0%)

Perfil: Default

Preparado para cargar o capturar

4:51 p. m. 28/10/2023

© 2011 Pearson Education, Inc. All rights reserved. This publication is protected by copyright. Any unauthorized distribution or reproduction of this work is illegal. All other rights reserved.

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

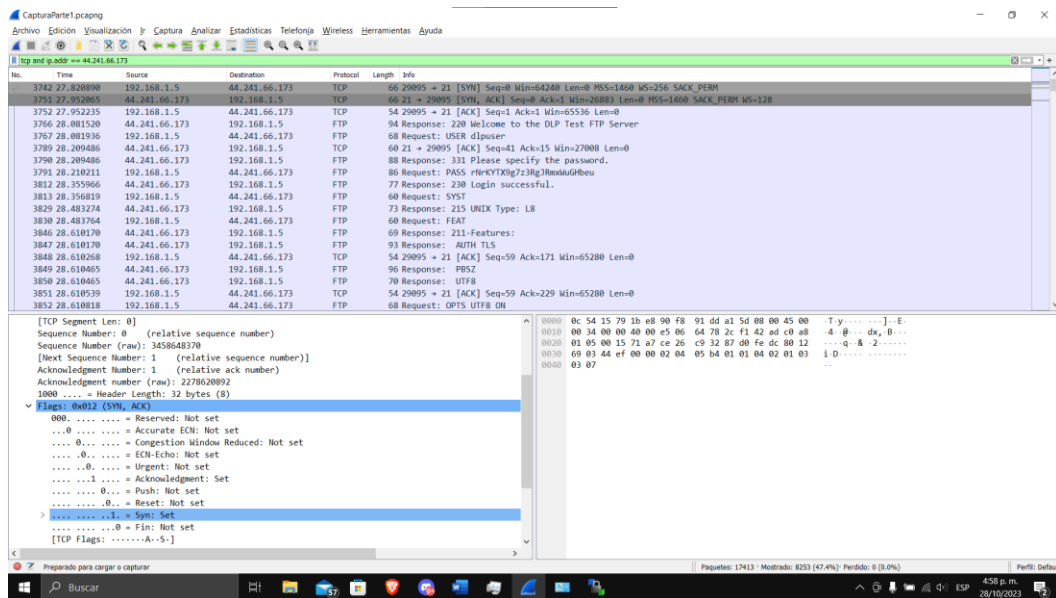
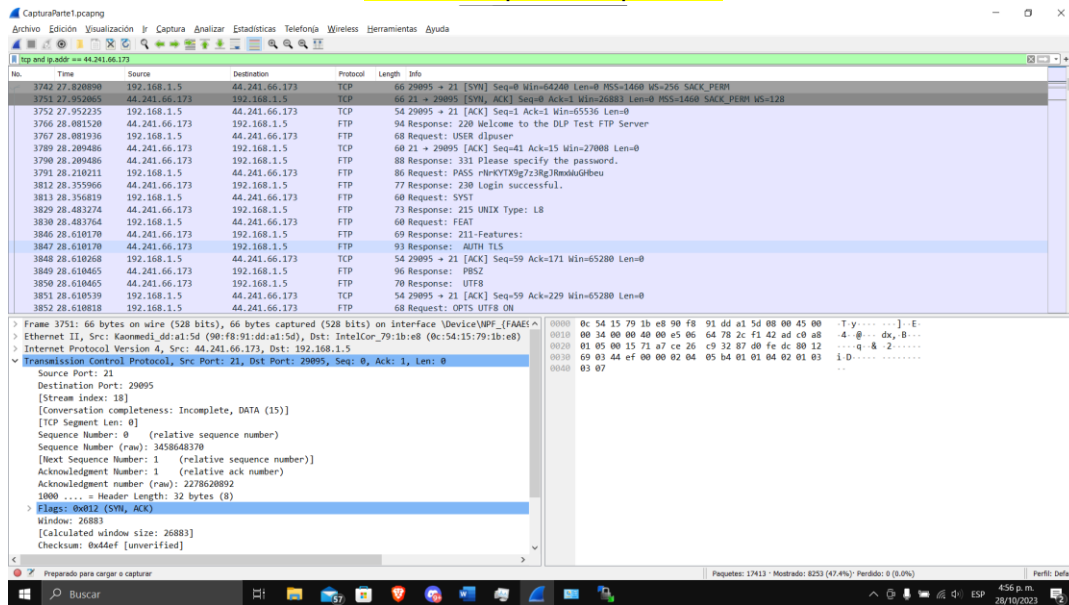
Dirección IP de origen	44.241.66.173
Dirección IP de destino	192.168.1.5
Número de puerto de origen	21
Número de puerto de destino	29095
Número de secuencia	0 relative sequence number
Número de reconocimiento	2278620892 raw
Longitud del encabezado	32 bytes
Tamaño de la ventana	26883

En la etapa final de la negociación para establecer las comunicaciones, la PC envía un mensaje de reconocimiento al servidor. Observe que solo el bit ACK está establecido en 1, y 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 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]
  
```

Colocar captura de pantalla



Complete la siguiente información sobre el mensaje de ACK.

Dirección IP de origen	192.168.1.5
Dirección IP de destino	44.241.66.173
Número de puerto de origen	29095
Número de puerto de destino	21
Número de secuencia	1 relative sequence number
Número de reconocimiento	3458648371
Longitud del encabezado	20 bytes
Tamaño de la ventana	65536

¿Cuántos otros datagramas TCP contenían un bit SYN?

2

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

Colocar captura de pantalla

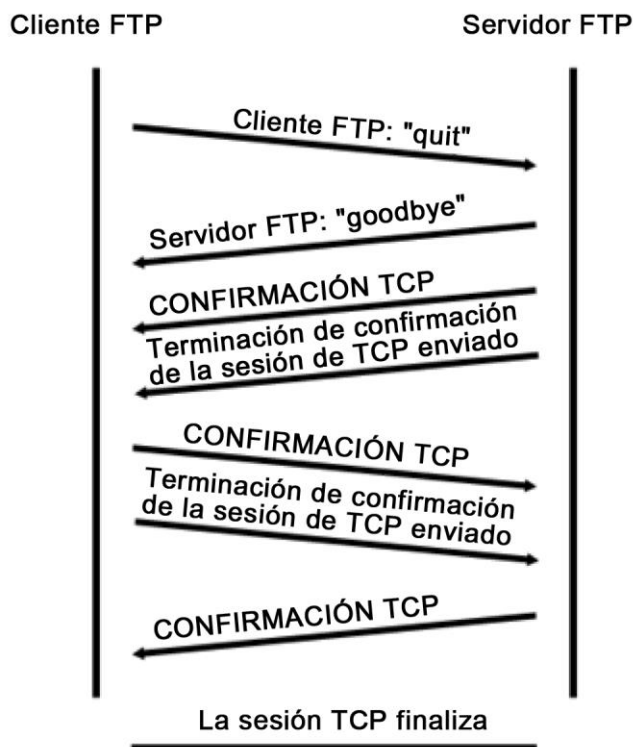
Práctica de laboratorio: Uso de Wireshark para examinar capturas de TCP y UDP

Wireshark capture of an FTP session. The packet list shows a sequence of FTP commands and responses. Packet 3752 is selected, showing the details of the first data packet (2278620892 bytes). The packet details pane shows the TCP segment structure, including sequence numbers, acknowledgment numbers, and flags (ACK). The packet bytes pane shows the raw data in hexadecimal and ASCII.

Wireshark capture of an FTP session. The packet list shows a sequence of FTP commands and responses. Packet 3752 is selected, showing the details of the first data packet (2278620892 bytes). The packet details pane shows the TCP segment structure, including sequence numbers, acknowledgment numbers, and flags (ACK). The packet bytes pane shows the raw data in hexadecimal and ASCII.

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.

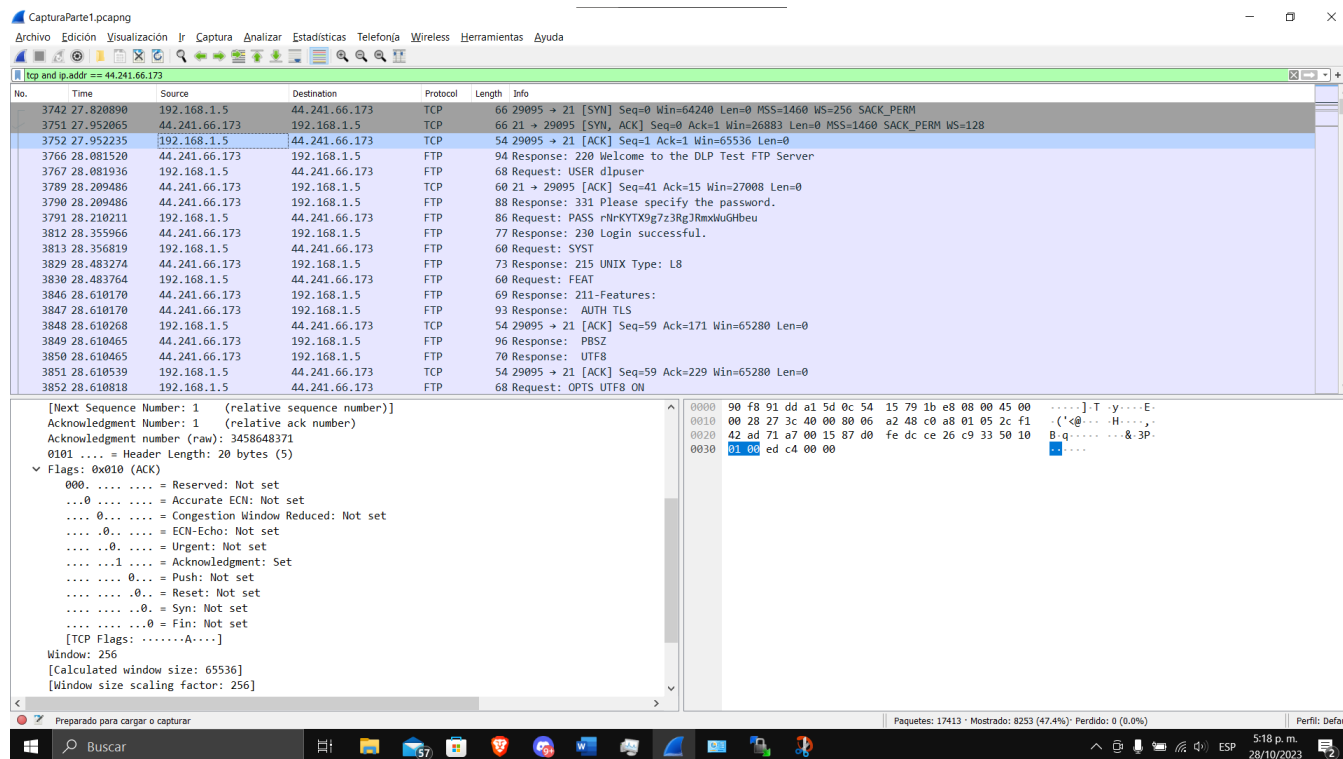


Si se aplica un filtro **ftp**, puede examinarse la secuencia completa del tráfico FTP en Wireshark. Observe la secuencia de los eventos durante esta sesión FTP. Para recuperar el archivo, se utilizó el nombre de usuario **dlpuser**. Una vez que se completó la transferencia de archivos, el usuario finalizó 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, ser
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
54	17.354538000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT cor
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 conn
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
82	25.136886000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT cor
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 conn
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.

Práctica de laboratorio: Uso de Wireshark para examinar capturas de TCP y UDP

Colocar captura de pantalla



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 de dúplex completo, 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 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, la sesión TCP terminó entre el servidor FTP y la PC.

147	30.48299200	(192.168.1.17)	198.246.117.106	FTP	60 Request: QUIT
148	30.56511700	(198.246.117.106)	192.168.1.17	FTP	68 Response: 221 Goodbye.
149	30.56646700	(198.246.117.106)	192.168.1.17	TCP	54 21→49411 [FIN, ACK] Seq=325 Ack=99 win=1
150	30.56653200	(192.168.1.17)	198.246.117.106	TCP	54 49411→21 [ACK] Seq=99 Ack=326 win=7868 L
151	30.56679900	(192.168.1.17)	198.246.117.106	TCP	54 49411→21 [FIN, ACK] Seq=99 Ack=326 win=7
152	30.66777000	(198.246.117.106)	192.168.1.17	TCP	54 21→49411 [ACK] Seq=326 Ack=100 win=13209

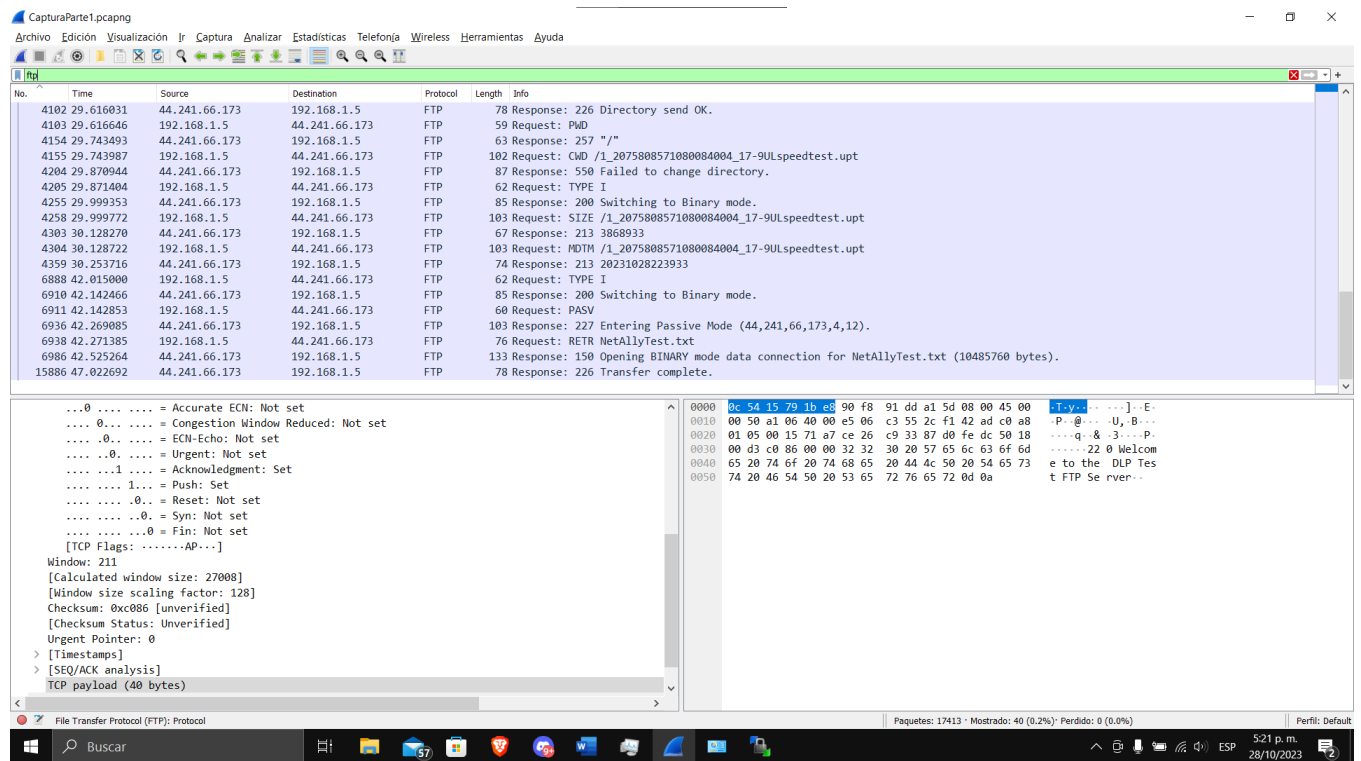
Frame 149: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

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: 325, Ack: 99, Len: 0

Colocar captura de pantalla



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



- Establezca una conexión de consola y Ethernet entre PC-A y S1.
- Configure manualmente la dirección IP en la PC a 192.168.1.3. No se requiere configurar el gateway predeterminado.
- Configure el switch. Asigne la dirección IP 192.168.5.1 a VLAN 1. Verifique la conectividad con la PC haciendo ping a 192.168.1.3. Resuelva cualquier problema que se presente. – cada pareja un segmento de red

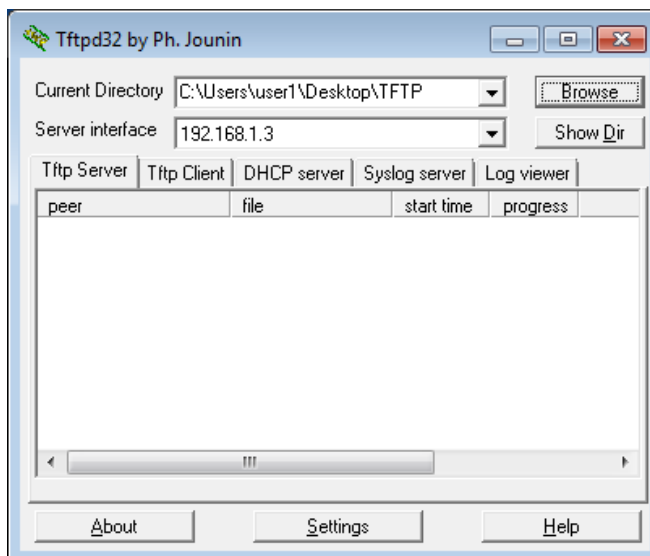
```
Switch> enable
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.5.1 255.255.255.0
S1(config-if)# no shut
*Mar  1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar  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
d. Guarde la configuración en ejecución en la NVRAM.
S1# copy run start
Ó
S1# write memory
```

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 en esta ubicación.
- Inicie **tftpd32** en la PC.

- c. Haga clic en **Browse** (Examinar) y cambie el directorio actual a **C:\Users\user1\Desktop\TFTP** reemplazando user1 por su nombre de usuario.

El servidor TFTP debería verse así:



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

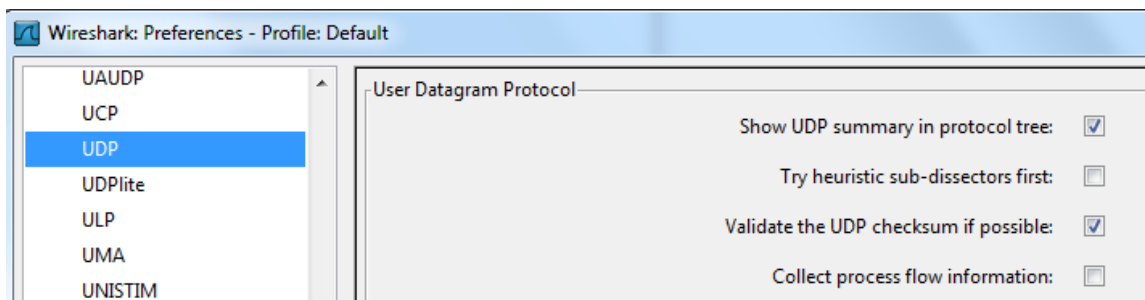
- d. Pruebe la capacidad de copiar un archivo del switch a la PC con TFTP. Resuelva cualquier problema que se presente.

```
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 se copió el archivo, está listo para ir al paso siguiente. Si el archivo no se copió, resuelva los problemas que se presenten. Si recibe el mensaje de error **%Error opening tftp (Permission denied)** (Error al abrir tftp [permiso denegado]), determine si el firewall está bloqueando el protocolo TFTP y si está copiando a una ubicación donde su nombre de usuario tiene el permiso adecuado, como el escritorio.

Paso 3: Capturar una sesión de TFTP en Wireshark

- a. Abra Wireshark. En el menú **Edit** (Editar), seleccione **Preferences** (Preferencias) y haga clic en el signo (+) 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).

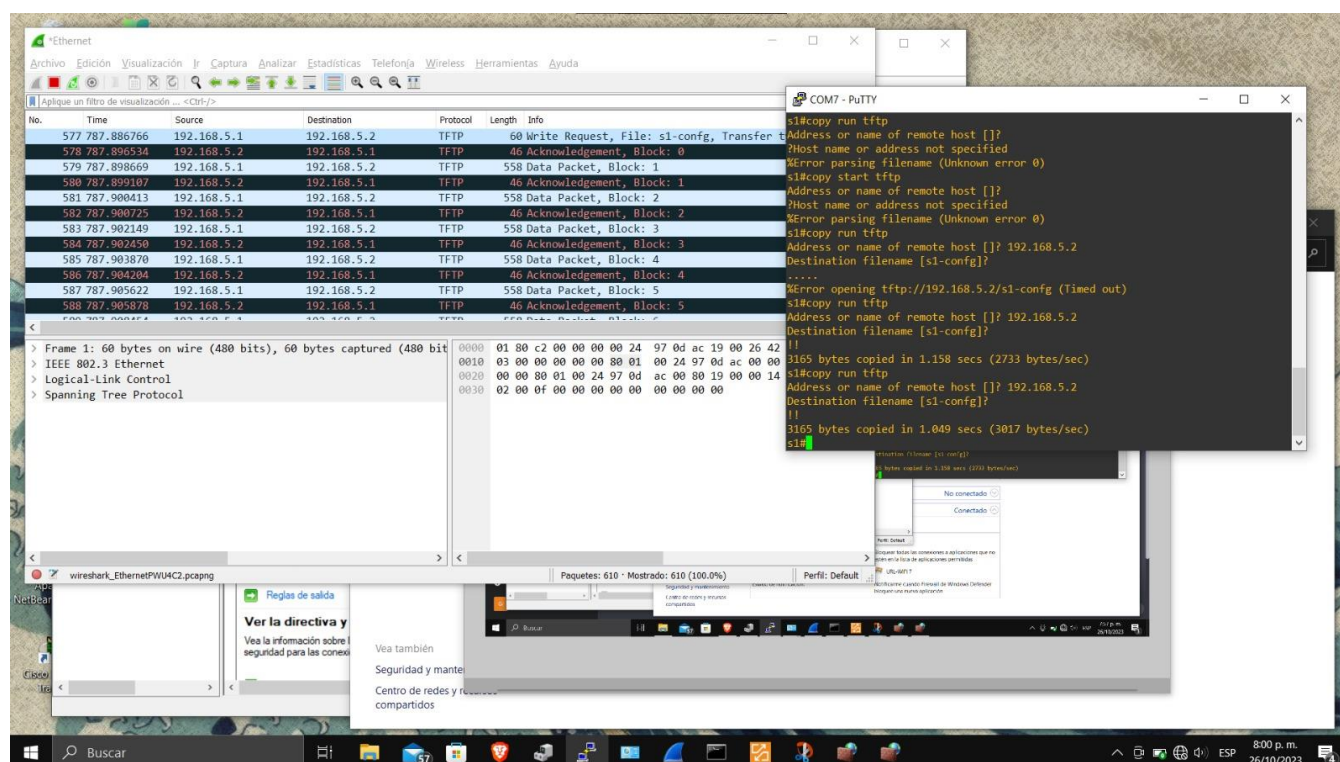


- Inicio una captura de Wireshark.
- Ejecute el comando **copy start tftp** en el switch.
- Detenga la captura de Wireshark.

Filter: **tftp** Expression... Clear Apply Save

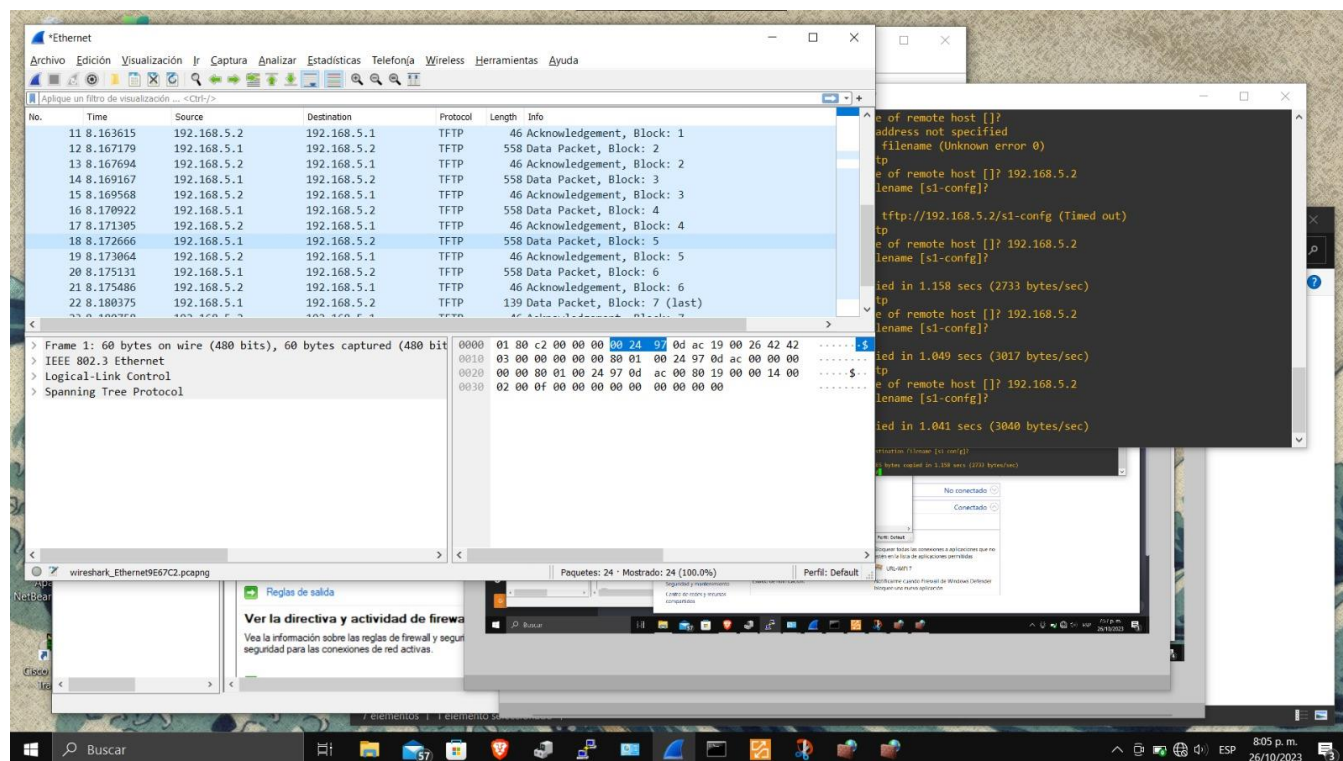
No.	Time	Source	Destination	Protocol	Length	Info
12	9.75564700	192.168.1.1	192.168.1.3	TFTP	60	Write Request, File: sl-config, Transfer type: octet
13	9.75668700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 0
14	9.75794800	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 1
15	9.75804400	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 1
16	9.75905100	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 2
17	9.75911700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 2
18	9.76013200	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 3
19	9.76018700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 3
20	9.76227300	192.168.1.1	192.168.1.3	TFTP	148	Data Packet, Block: 4 (last)
21	9.76240000	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 4

Colocar captura de pantalla



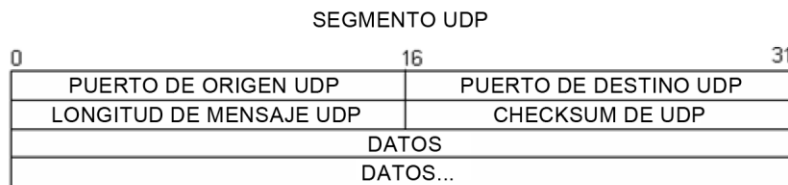
- e. Configure el filtro en **tftp**. El resultado debe ser similar al que se muestra más arriba. Esta transferencia de TFTP se utiliza para analizar las operaciones de UDP de la capa de transporte.

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.



Encabezado UDP	<ul style="list-style-type: none"> <ul style="list-style-type: none"> User Datagram Protocol, Src Port: 62513 (62513), Dst Port: tftp (69) Source port: 62513 (62513) Destination port: tftp (69) Length: 25 Checksum: 0x482c [correct]
Datos UDP	<ul style="list-style-type: none"> <ul style="list-style-type: none"> Trivial File Transfer Protocol [DESTINATION File: s1-config] Opcode: Write Request (2) DESTINATION File: s1-config Type: octet

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:

Dirección IP de origen	192.168.5.1
Dirección IP de destino	192.168.5.2
Número de puerto de origen	63000
Número de puerto de destino	69
Longitud del mensaje UDP	25
Checksum de UDP	0x3E46

¿Cómo verifica UDP la integridad del datagrama?

Este protocolo UDP lo realiza mediante el campo de checksum el cual tiene como objetivo verificar la integridad del datagrama. Este checksum que se menciona se calcula usando el contenido del datagrama y algunas partes de la cabecera IP. Este datagrama se modifica en la fase la transmisión, el cálculo del checksum en el extremo receptor no coincidirá con el valor del checksum enviado, indicando un error.

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

Dirección IP de origen	192.168.5.2
Dirección IP de destino	192.168.5.1
Número de puerto de origen	65389
Número de puerto de destino	63000
Longitud del mensaje UDP	12
Checksum de UDP	0x8b71 checksum es incorrecto debería ser 0x7Ef7

- ☒ 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, 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".

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?

Estos protocolos distintos con diferentes enfoques para manejar la comunicación en la red. TCP un protocolo orientado a la conexión, este establece una conexión firme antes de la transferencia de datos, lo que ganamos con esto es la entrega de paquetes y manteniendo su secuencia. Por otro lado el UDP es un protocolo sin conexión que envía paquetes sin establecer una conexión previa y no garantiza su entrega ni el orden, este tiende a ser más ligero y rápido.

Desafío

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

Colocar capturas de pantalla donde se identifique el usuario y la contraseña dentro de Wireshark

3767	28.081936	192.168.1.5	44.241.66.173	FTP	68 Request: USER dlpuser
3789	28.209486	44.241.66.173	192.168.1.5	TCP	60 21 → 29095 [ACK] Seq=41 Ack=15 Win=27008 Len=0
3790	28.209486	44.241.66.173	192.168.1.5	FTP	88 Response: 331 Please specify the password.
3791	28.210211	192.168.1.5	44.241.66.173	FTP	86 Request: PASS rNrKYTX9g7z3RgJRmxWuGHbeu

Limpieza

Salvo que el instructor indique lo contrario:

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

Configuraciones de dispositivos

Colocar captura de pantalla del running-configuration del switch

Entregables en el portal por pareja:

- 1) Manual de laboratorio resuelto (PDF)
- 2) Archivos de captura de Wireshark (Parte 1 y Parte 2)
- 3) Archivo de backup del Switch Cisco Catalyst 2960