Solución

Laboratorio 4.5.2: Protocolos de la capa de transporte TCP/IP, TCP y UDP

Diagrama de topología

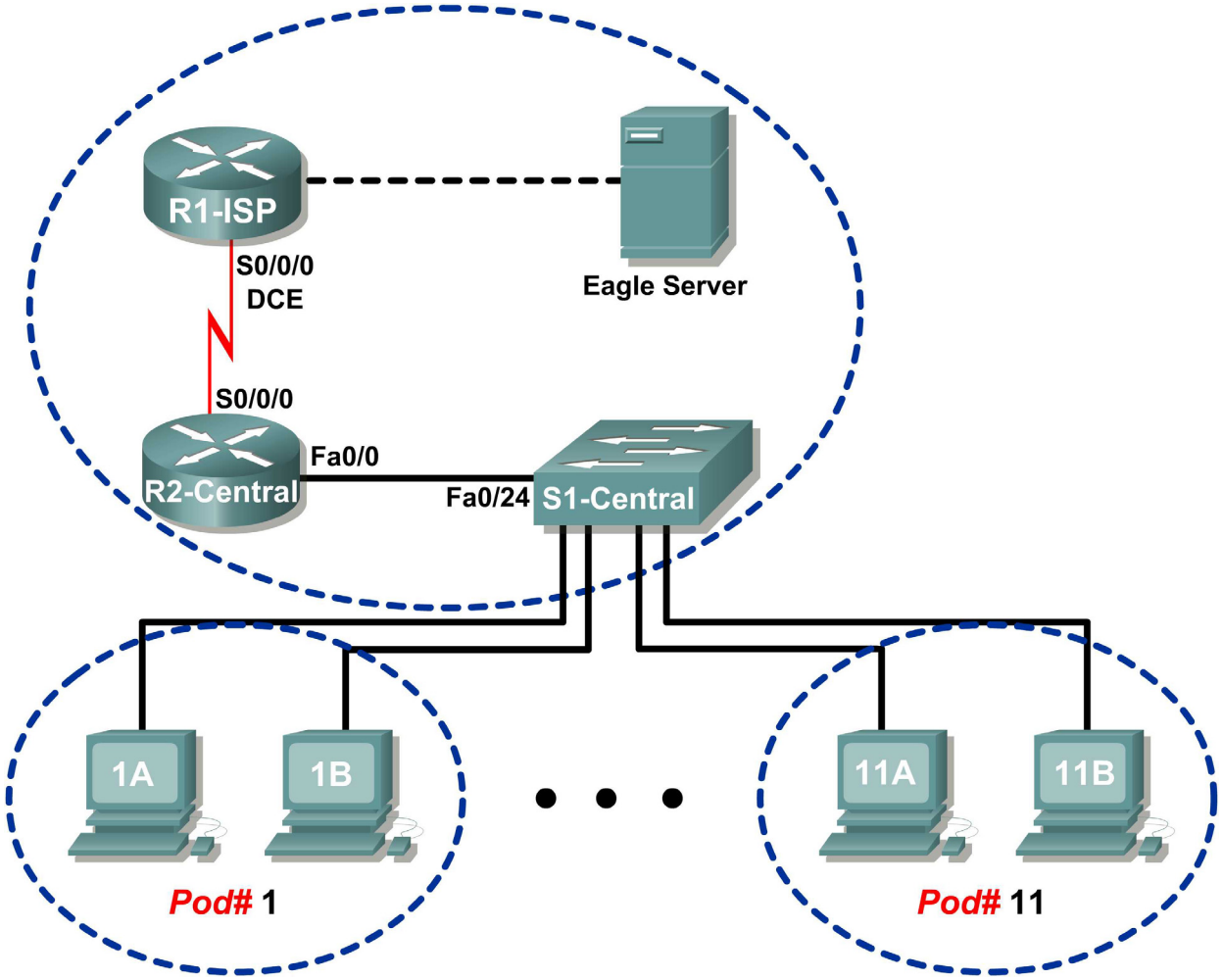

Tabla de direccionamiento

Dispositivo	Interfaz	Dirección IP	Máscara de subred	Gateway por defecto
R1-ISP	S0/0/0	10.10.10.6	255.255.255.252	No aplicable
	Fa0/0	192.168.254.253	255.255.255.0	No aplicable
R2-Central	S0/0/0	10.10.10.5	255.255.255.252	No aplicable
	Fa0/0	172.16.255.254	255.255.0.0	No aplicable
Eagle Server	No aplicable	192.168.254.254	255.255.255.0	192.168.254.253
	No aplicable	172.31.24.254	255.255.255.0	No aplicable
host*Pod*#A	No aplicable	172.16.*Pod*#.1	255.255.0.0	172.16.255.254
host*Pod*#B	No aplicable	172.16.*Pod*#.2	255.255.0.0	172.16.255.254
S1-Central	No aplicable	172.16.254.1	255.255.0.0	172.16.255.254

Objetivos de aprendizaje

- Identificar campos de encabezado y operación TCP mediante el uso de una captura de sesión FTP Wireshark.
- Identificar campos de encabezado y operación UDP mediante el uso de una captura de sesión TFTP Wireshark.

Información básica

Los dos protocolos en la capa de Transporte TCP/IP son: el Transmission Control Protocol (TCP) definido en RFC 761, en enero de 1980; y el User Datagram Protocol (UDP), definido en RFC 768, en agosto de 1980. Ambos protocolos admiten la comunicación de protocolo de capa superior. Por ejemplo, el TCP se utiliza para proveer soporte de la capa de Transporte para los protocolos HTTP y FTP, entre otros. El UDP provee soporte de la capa de Transporte para servicios de nombres de dominio (DNS) y Trivial File Transfer Protocol (TFTP), entre otros.

La capacidad para entender las partes de los encabezados y de la operación TCP y UDP es una habilidad muy importante para los ingenieros de red.

Escenario

Mediante la captura Wireshark, analizar los campos de encabezado del protocolo UDP y TCP para la transferencia de archivos entre el equipo host y Eagle Server. Si no se cargó Wireshark en el equipo host del módulo, lo puede descargar desde `ftp://eagle-server.example.com/pub/eagle_labs/eagle1/chapter4/`, archivo `wireshark-setup-0.99.4.exe`.

Las utilidades de Windows de línea de comandos `ftp` y `tftp` se utilizará para conectarse a Eagle Server y descargar archivos. (No esta activo tendreis que utilizar otro servidor ftp y tftp) **(en mi caso utilizo un servidor de ftp y tftp, en mi red local con ip 172.16.20.1)**

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

Paso 1: Capture una sesión FTP.

Las sesiones TCP se controlan y administran debidamente con información que se intercambia en los campos de encabezado TCP. En esta tarea se realizará una sesión FTP con Eagle Server. Cuando finalice, se analizará la captura de sesión. Las computadoras con Windows utilizan al cliente FTP, **ftp**, para conectarse al servidor FTP. Una ventana de línea de comandos iniciará la sesión FTP y se descargar el archivo de configuración de texto para S1 central de Eagle Server, `/pub/eagle_labs/eagle1/chapter4/s1-central` al equipo host.

Abra una ventana de línea de comandos con un clic en Iniciar / Ejecutar, escriba cmd y luego presione Aceptar.

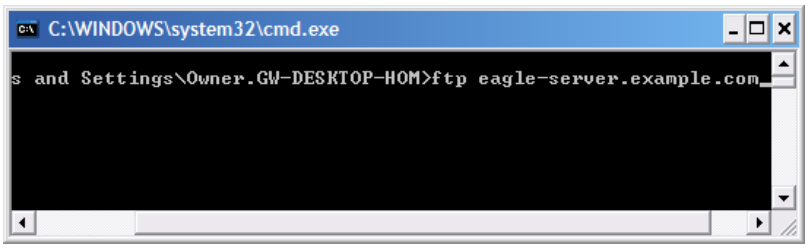

Figura 1. Ventana de línea de comandos.

Deberá abrirse una ventana similar a la Figura 1.

Inicie una captura Wireshark en la interfaz que tenga la dirección IP `172,16.Pod#.[1-2]`.

Inicie una conexión FTP con Eagle Server. Escriba el comando:

```
> ftp eagle-server.example.com (hay que utilizar otro servidor ftp)
```

Cuando se le pida un nombre de usuario, escriba **anonymous.** Cuando se le pida una contraseña, presione **<INTRO>.**

Cambie el directorio FTP a `/pub/eagle_labs/eagle1/chapter4/`:

```
ftp> cd /pub/eagle_labs/eagle1/chapter4/
```

Descargue el archivo `s1-central`:

```
ftp> get s1-central
```

Cuando termine, finalice las sesiones FTP en cada ventana de línea de comandos con el comando FTP **quit:**

```
ftp> quit
```

Cierre la ventana de línea de comandos con el comando **exit:**

```
> exit
```

Detenga la captura Wireshark.

Paso 2: Analice los campos TCP.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.1	192.168.254.254	TCP	1052 > ftp [SYN] Seq=0 Len=0 MSS=1460
2	0.000568	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
3	0.000610	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.004818	192.168.254.254	172.16.1.1	FTP	Response: 220 Welcome to the eagle-server FTP service.
5	0.115430	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=1 Ack=47 Win=64194 Len=0
6	8.223541	172.16.1.1	192.168.254.254	FTP	Request: USER anonymous
7	8.224089	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [ACK] Seq=47 Ack=17 Win=5840 Len=0
8	8.224126	192.168.254.254	172.16.1.1	FTP	Response: 331 Please specify the password.
9	8.327214	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=17 Ack=81 Win=64160 Len=0
10	9.517629	172.16.1.1	192.168.254.254	FTP	Request: PASS
11	9.519135	192.168.254.254	172.16.1.1	FTP	Response: 230 Login successful.
12	9.629097	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=24 Ack=104 Win=64137 Len=0
13	32.365752	172.16.1.1	192.168.254.254	FTP	Request: CWD /pub/eagle_labs/eagle1/chapter4
14	32.366375	192.168.254.254	172.16.1.1	FTP	Response: 250 Directory successfully changed.
15	32.376653	172.16.1.1	192.168.254.254	FTP	Request: PORT 172,16,1,1,4,33
16	32.377165	192.168.254.254	172.16.1.1	FTP	Response: 200 PORT command successful. Consider using PASV.
17	32.381726	172.16.1.1	192.168.254.254	FTP	Request: RETR s1-central
18	32.382337	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [SYN] Seq=0 Len=0 MSS=1460 TSV=4755496 TSER=0 WS=2
19	32.382398	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=0 TSV=0 TSER=0
20	32.382777	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=4755496 TSER=0
21	32.382891	192.168.254.254	172.16.1.1	FTP	Response: 150 Opening BINARY mode data connection for s1-central (3100 bytes).
22	32.383528	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 1448 bytes
23	32.383589	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 1448 bytes
24	32.383631	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [ACK] Seq=1 Ack=2897 Win=64240 Len=0 TSV=36854 TSER=4755496
25	32.383736	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 204 bytes
26	32.383753	192.168.254.254	172.16.1.1	FTP	Response: 226 File send OK.
27	32.383773	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=100 Ack=281 Win=63960 Len=0
28	32.383779	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [FIN, ACK] Seq=3101 Ack=1 Win=5840 Len=0 TSV=4755496 TSER=0
29	32.383805	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [ACK] Seq=1 Ack=3102 Win=64036 Len=0 TSV=36854 TSER=4755496
30	32.389457	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [FIN, ACK] Seq=1 Ack=3102 Win=64036 Len=0 TSV=36854 TSER=4755496
31	32.389845	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [ACK] Seq=3102 Ack=2 Win=5840 Len=0 TSV=4755503 TSER=36854
32	34.438952	172.16.1.1	192.168.254.254	FTP	Request: QUIT
33	34.439532	192.168.254.254	172.16.1.1	FTP	Response: 221 Goodbye.
34	34.439893	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [FIN, ACK] Seq=295 Ack=106 Win=5840 Len=0
35	34.439934	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=106 Ack=296 Win=63946 Len=0
36	34.442705	172.16.1.1	192.168.254.254	TCP	1052 > ftp [FIN, ACK] Seq=106 Ack=296 Win=63946 Len=0
37	34.443144	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [ACK] Seq=296 Ack=107 Win=5840 Len=0

Figura 2. Captura FTP.

Cambie a las ventanas de captura Wireshark. La ventana superior contiene resumen de información para cada registro capturado. La captura realizada por el estudiante debe ser similar a la captura que se muestra en la Figura 2. Antes de profundizar en los detalles del paquete TCP, se necesita una explicación del resumen de información. Cuando el cliente FTP está conectado al servidor FTP, el protocolo TCP de la capa de Transporte creó una sesión confiable. 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. Finalmente, cuando la sesión FTP finaliza, TCP realiza un cierre y terminación ordenados.

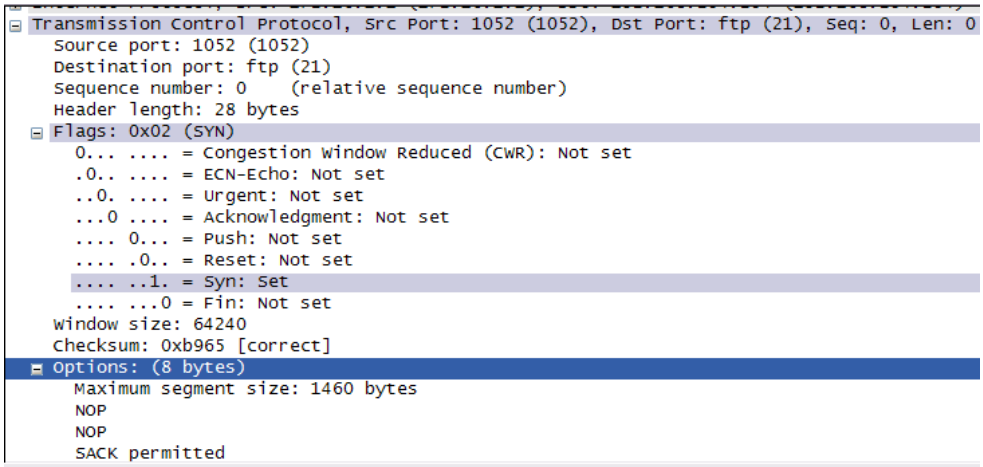

Figura 3. Captura Wireshark de un datagrama TCP.

Hay información TCP detallada disponible en la ventana del medio, en Wireshark. Resalte el primer datagrama TCP del equipo host y mueva el puntero del mouse hacia la ventana del medio. Puede ser necesario ajustar la ventana del medio y expandir el registro TCP con un clic en la casilla de expansión de protocolo. El datagrama TCP expandido debe ser similar a la Figura 3.

¿Cómo se identifica el primer datagrama en una sesión TCP?

Se identifica enviando la sequence number 0 y en los flags pone a 1 el bit Syn

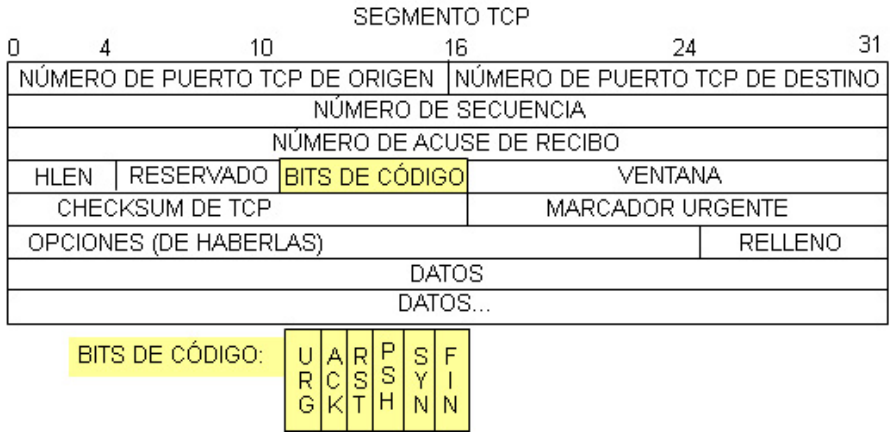

Figura 4. Campos del paquete TCP.

Observe la Figura 4, un diagrama de datagrama TCP. Se provee a los estudiantes una explicación de cada campo para refrescarles la memoria:

- **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 1023.
- **El número de puerto de destino** se utiliza para identificar el protocolo de capa superior o la aplicación en un sitio remoto. Los valores dentro del intervalo 0 – 1023 representan a los llamados "puertos bien conocidos" y están asociados con servicios y aplicaciones conocidos (como se describe en RFC 1700, telnet, File Transfer Protocol (FTP), HyperText Transfer Protocol (HTTP), etc.). La combinación de campo cuádruple (dirección IP de origen, puerto de origen, dirección IP de destino, puerto de destino) identifica de manera exclusiva la sesión, tanto del emisor como del receptor.
- **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, configurar sólo cuando una sesión TCP nueva se negocia durante un protocolo de enlace de tres vías).
    - FIN (Finalizar, solicitud para cerrar la sesión TCP).
- **El tamaño de la ventana** es el valor de la ventana deslizante; cuántos octetos se pueden enviar antes de esperar un acuse de recibo.
- **El puntero urgente** se utiliza sólo con un señalizador URG (Urgente) cuando el emisor necesita enviar datos urgentes al receptor.
- **Opciones:** La única opción definida actualmente es el tamaño de segmento TCP máximo (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.

Del equipo host del módulo a Eagle Server (sólo el bit SYN se fija en 1):

Dirección IP de origen: 172.16.___.___	172.16.186.85
Dirección IP destino: _____	172.16.20.1
Número de puerto de origen: _____	1923
Número de puerto de destino: _____	21
Número de secuencia: _____	0
Número de acuse de recibo: _____	(no existe)
Longitud del encabezado: _____	28 bytes
Tamaño de la ventana: _____	65535 bytes

De Eagle Server al equipo host del módulo (sólo los bits SYN y ACK se fijan en 1):

Dirección IP de origen: _____	172.16.20.1
Dirección IP destino: 172.16.___.___	172.16.186.85
Número de puerto de origen: _____	21
Número de puerto de destino: _____	1923
Número de secuencia: _____	0
Número de acuse de recibo: _____	1
Longitud del encabezado: _____	28 bytes
Tamaño de la ventana: _____	14600 bytes

Del equipo host del módulo a Eagle Server (sólo el bit ACK se fija en 1):

Dirección IP de origen: 172.16.___.___	172.16.186.85
Dirección IP destino: _____	172.16.20.1
Número de puerto de origen: _____	1923
Número de puerto de destino: _____	21
Número de secuencia: _____	1
Número de acuse de recibo: _____	1
Longitud del encabezado: _____	20 bytes
Tamaño de la ventana: _____	65535 bytes

A excepción de la sesión TCP iniciada cuando se realizó una transferencia de datos, ¿cuántos otros datagramas TCP contienen un bit SYN?

Se vuelve a activar el bit Syn cuando el servidor va a iniciar la descarga del archivo que solicitamos con la orden get, y justo en la siguiente trama, cuando nuestro host le responde.(en el caso de que hubieramos subido un archivo con la orden put, al servidor se hubiera activado tambien pero en orden inverso)

Los atacantes se aprovechan del protocolo de enlace de tres vías al iniciar una conexión “half-open”. En esta secuencia la sesión TCP inicial envía un datagrama TCP con el bit SYN establecido y el receptor envía un datagrama TCP relacionado con los bits SYN ACK establecidos. Un bit ACK final no se envía nunca para finalizar el intercambio TCP. En cambio, se inicia una conexión TCP nueva de manera half-open. Con suficientes sesiones TCP en estado half-open, el equipo receptor agotará recursos y colapsará. Un colapso puede incluir una pérdida de servicios de red o un daño en el sistema operativo. De cualquier modo, el atacante gana. El servicio de red se ha detenido en el receptor. Éste es un ejemplo de ataque de denegación de servicio (DoS).

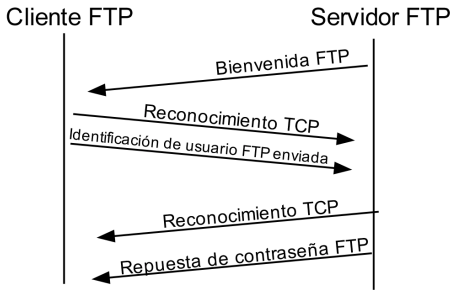

Figura 5. Administración de sesión TCP.

El cliente y el servidor FTP se comunican uno con el otro sin saber y sin importarles que TCP tenga el control y manejo de la sesión. Cuando el servidor FTP envía una Respuesta: 220 al cliente FTP, la sesión TCP del cliente FTP envía un acuse de recibo a la sesión TCP en Eagle Server. Esta secuencia se muestra en la Figura 5 y es visible en la captura Wireshark.

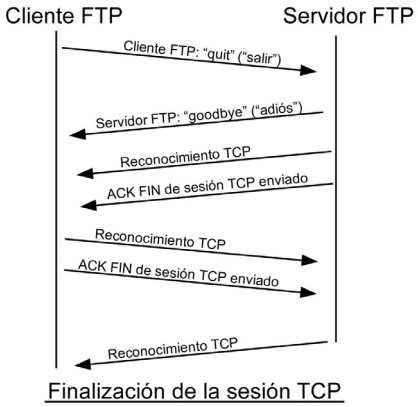

Figura 6. Terminación de la sesión TCP ordenada.

Cuando la sesión FTP terminó, el cliente FTP envía un comando para "`salir`". El servidor FTP acusa recibo de la terminación FTP con una `Respuesta 221 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 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 muestra en la Figura 6 y es visible en la captura Wireshark.

Sin una terminación ordenada, como por ejemplo cuando se interrumpe la conexión, las sesiones TCP esperarán un cierto período de tiempo hasta cerrarse. El valor de límite de tiempo de espera predeterminado varía, pero normalmente es de 5 minutos.

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

Paso 1: Capture una sesión TFTP.

Siga el procedimiento de la Tarea 1 de arriba y abra una ventana de línea de comandos. El comando TFTP tiene una sintaxis diferente a la de FTP. Por ejemplo: no hay autenticación. También, hay sólo dos comandos: **`get`**, para recuperar un archivo y **`put`**, para enviar un archivo.

```
>tftp -help

Transfiere los archivos a y desde un equipo remoto con el servicio TFTP en
funcionamiento.

TFTP [-i] host [GET | PUT] origen [destino]

        -i       Especifica el modo de transferencia binario (llamado también
                 octeto). En modo binario el archivo se transfiere
                 literalmente, byte a byte. Use este modo cuando
                 transfiera archivos binarios.
        host     Especifica el host remoto o local.
        GET      Transfiere el archivo destino en el host remoto al
                 archivo origen en el host local.
        PUT      Transfiere el archivo origen en el host local al
                 archivo destino en el host remoto.
        origen   Especifica el archivo a transferir.
      destino    Especifica dónde transferir el archivo.
```

Tabla 1. Sintaxis TFTP para un cliente TFTP Windows.

La Tabla 1 contiene sintaxis de cliente TFTP Windows. El servidor TFTP tiene su propio directorio en Eagle Server, `/tftpboot`, que es diferente de la estructura del directorio admitido por el servidor FTP. No se admite ninguna autenticación.

Inicie una captura Wireshark, luego descargue el archivo de configuración `s1-central` de Eagle Server con el cliente TFTP Windows. El comando y la sintaxis para realizar esto se muestran debajo:

```
>tftp eagle-server.example.com get s1-central(utilizamos el servidor anterior)
```

Paso 2: Analice los campos UDP.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.1	192.168.254.254	TFTP	Read Request, File: s1-central, Transfer type: netascii
2	0.003171	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 1
3	0.003314	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 1
4	0.003962	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 2
5	0.004021	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 2
6	0.004615	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 3
7	0.004673	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 3
8	0.005274	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 4
9	0.005332	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 4
10	0.005930	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 5
11	0.005989	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 5
12	0.006588	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 6
13	0.006644	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 6
14	0.007078	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 7 (last)
15	0.007131	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 7

Figura 7. Captura de resumen de una sesión UDP.

Cambie a las ventanas de captura Wireshark. La captura realizada por el estudiante debe ser similar a la captura que se muestra en la Figura 7. Se utilizará una transferencia TFTP para analizar la operación de capa de Transporte UDP.

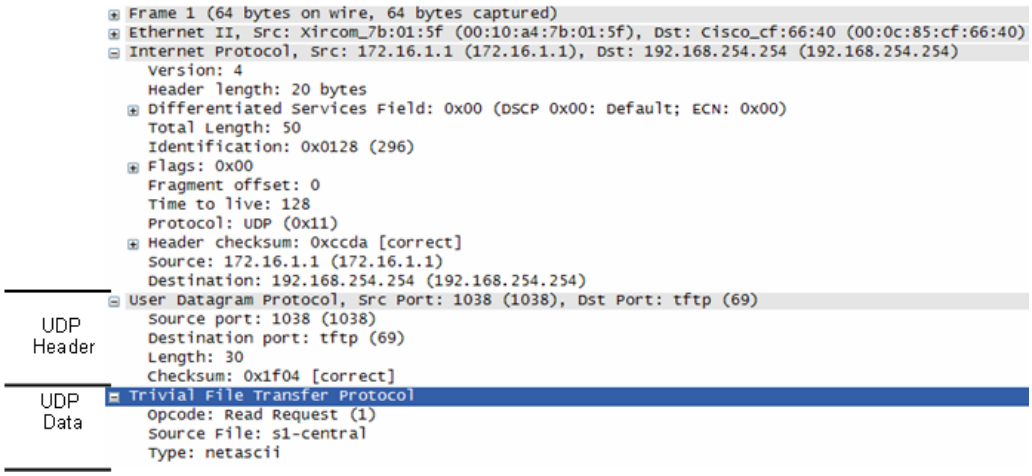

Figura 8. Captura Wireshark de un datagrama UDP.

Hay información UDP detallada disponible en la ventana del medio en Wireshark. Resalte el primer datagrama UDP del equipo host y mueva el puntero del mouse hacia la ventana del medio. Puede ser necesario ajustar la ventana del medio y expandir el registro UDP con un clic en la casilla de expansión de protocolo. El datagrama UDP expandido debe ser similar a la Figura 8.

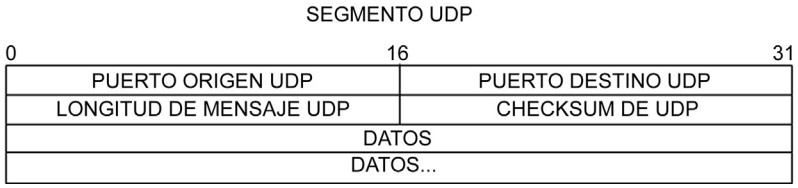

Figura 9. Formato UDP.

Observe la Figura 9, un diagrama de datagrama UDP. La información del encabezado está dispersa comparada con la del datagrama TCP. Sin embargo hay similitudes. Cada datagrama UDP es identificado por 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: 172.16.___.___	172.16.186.85
Dirección IP destino: _____	172.16.20.1
Número de puerto de origen: _____	1383
Número de puerto de destino: _____	69
Longitud de mensaje UDP: _____	25
Checksum de UDP: _____	0x922b [validation disabled]

¿Cómo verifica UDP la integridad del datagrama?

Tal como indica el Checksum, No la verifica. Se encargara la aplicación de la capa superior

Examine el primer paquete devuelto por Eagle Server. Complete la información acerca del encabezado UDP:

Dirección IP de origen:	172.16.20.1
Dirección IP destino: 172.16.___.___	172.16.186.85
Número de puerto de origen: _____	51077
Número de puerto de destino: _____	1383
Longitud de mensaje UDP: _____	27
Checksum de UDP: 0x_____	0x192e [validation disabled]

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.

Tarea 5: Reflexión

Esta práctica de laboratorio brindó a los estudiantes la oportunidad de analizar las operaciones de protocolo UDP y TCP de sesiones TFTP y FTP capturadas. TCP administra la comunicación de manera muy diferente a UDP, pero la confiabilidad y garantía ofrecidas requieren un control adicional sobre el canal de comunicación. UDP tiene menos sobrecarga y control, y el protocolo de capa superior debe proveer algún tipo de control de acuse de recibo. Sin embargo, ambos protocolos transportan datos entre clientes y servidores con el uso de los protocolos de la capa de Aplicación y son correctos para el protocolo de capa superior que cada uno admite.

Tarea 6: Desafío

Debido a que ni FTP ni TFTP son protocolos seguros, todos los datos transferidos se envían en texto sin cifrar. Esto incluye ID de usuario, contraseñas o contenidos de archivo en texto sin cifrar. Si analiza la sesión FTP de capa superior identificará rápidamente el id de usuario, contraseña y contraseñas de archivo 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 de configuración de id de usuario y contraseña.

Tarea 7: Limpieza

Durante esta práctica de laboratorio se transfirieron varios archivos al equipo host y se deben eliminar.

A menos que el instructor le indique lo contrario, apague las computadoras host. Llévese todo aquello que haya traído al laboratorio y deje el aula lista para la próxima clase.

