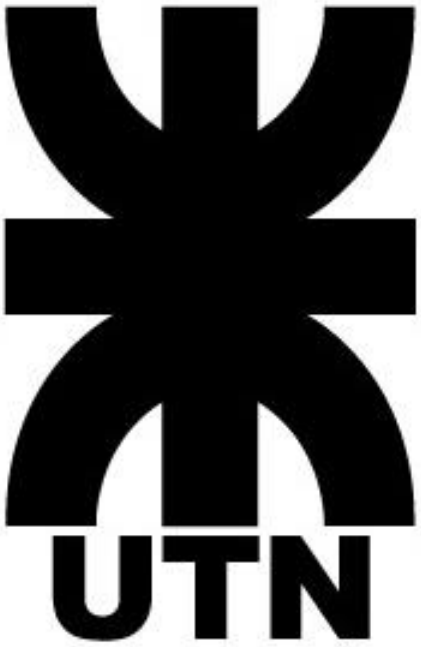
Universidad Tecnológica Nacional



Facultad Regional Delta

Laboratorio de redes de Información

2024

Trabajo Práctico N°3| Capa de Transporte

Alumno: Gonzalez, Tomas

Profesor: Carrizo Carlos

Contenido

Consignas

Temas: TCP/UDP, DNS, FTP.

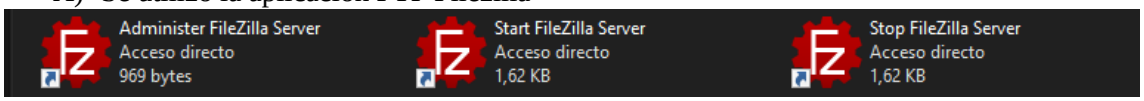
Herramientas a utilizar: PC con Windows, aplicación Wireshark, Servidor FTP.

1) Desde una PC con Windows:
   - Instale un servidor FTP Open Source.
   - Configure el servidor FTP para recibir conexiones entrantes mediante usuario y contraseña.
   - Inicie el monitoreo de la placa de red con Wireshark con el modo promiscuo desactivado.
   - Una vez iniciado el monitoreo utilice un cliente FTP y conéctese al servidor FTP desde la misma PC utilizando para la conexión la dirección IP: 127.0.0.1.
   - Suba un archivo con de texto con su nombre al servidor FTP.
   - Desconéctese y finalice el monitoreo.

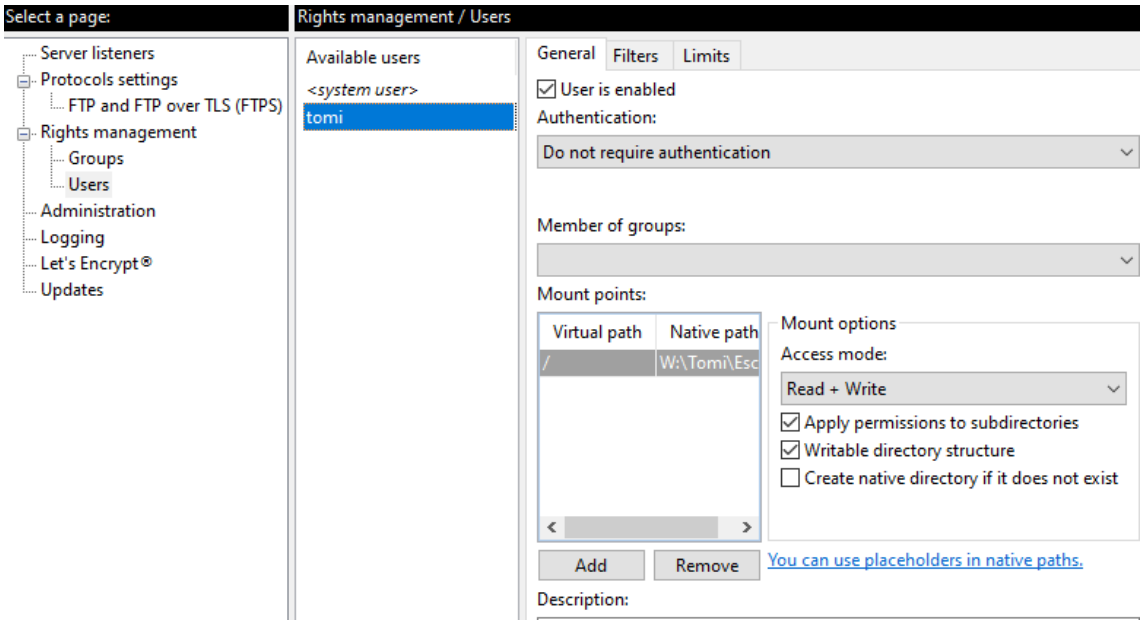
a) Mencione qué aplicación cliente y servidor FTP utilizó.
b) Analice los paquetes capturados de la conexión FTP, determine y documente los números de secuencia y el estado de los flags en la cabecera TCP de los 3 paquetes de inicio de la conexión.
c) Determine el puerto del servidor FTP y del cliente utilizado en la conexión.
d) Mencione una aplicación que utilice el protocolo UDP. Comente por qué se utiliza UDP en esta aplicación en vez de TCP.

Resolución

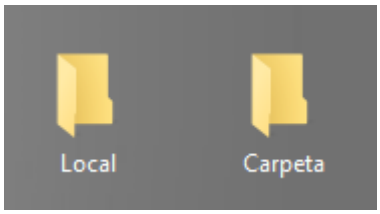
A) Se utilizo la aplicación FTP Filezilla



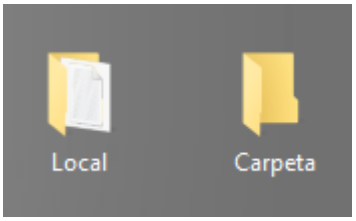
En la aplicación se configuro un usuario de nombre tomi y que tiene su carpeta asociada en el escritorio de la computadora



Se crearon dos carpetas en el escritorio, una llamada “Carpeta” la cual es la carpeta que utilizara filezilla como carpeta de servidor, y la carpeta “Local”, la cual sera la carpeta cliente que está loclamente en la computadora.



Luego se crea el archivo “tomi.txt” dentro de la carpeta local.



B) Iniciamos la captura con el wireshark, luego de eso, iniciamos la aplicación de FTP propia de windows desde la CMD y con el comando open nos conectamos a la IP del Local Host para establecer conexión en el mismo ordenador con el servidor Firezilla.

```
C:\Users\Tomi>ftp
ftp> open 127.0.0.1
Conectado a 127.0.0.1.
220-FileZilla Server 1.8.2
220 Please visit https://filezilla-project.org/
202 UTF8 mode is always enabled. No need to send this command
```

Luego de eso, iniciamos sesion con el usuario y contraseña que creamos en el Firezilla

```
C:\Users\Tomi>ftp
ftp> open 127.0.0.1
Conectado a 127.0.0.1.
220-FileZilla Server 1.8.2
220 Please visit https://filezilla-project.org/
202 UTF8 mode is always enabled. No need to send this command
Usuario (127.0.0.1:(none)): tomi
331 Please, specify the password.
Contraseña:
230 Login successful.
```

Luego establecemos como ruta de la carpeta local, la carpeta llamada “Local” que fue creada anteriormente.

```
C:\Users\Tomi>ftp
ftp> open 127.0.0.1
Conectado a 127.0.0.1.
220-FileZilla Server 1.8.2
220 Please visit https://filezilla-project.org/
202 UTF8 mode is always enabled. No need to send this command
Usuario (127.0.0.1:(none)): tomi
331 Please, specify the password.
Contraseña:
230 Login successful.
ftp> lcd W:\Tomi\Escritorio\Local
Directorio local ahora W:\Tomi\Escritorio\Local.
```

Por ultimo pasamos el archivo .txt contenido en la carpeta local a la carpeta del servidor y luego procedemos a salir con el comando quit

```
C:\Users\Tomi>ftp
ftp> open 127.0.0.1
Conectado a 127.0.0.1.
220-FileZilla Server 1.8.2
220 Please visit https://filezilla-project.org/
202 UTF8 mode is always enabled. No need to send this command
Usuario (127.0.0.1:(none)): tomi
331 Please, specify the password.
Contraseña:
230 Login successful.
ftp> lcd W:\Tomi\Escritorio\Local
Directorio local ahora W:\Tomi\Escritorio\Local.
ftp> put tomi.txt
200 PORT command successful.
150 Starting data transfer.
226 Operation successful
ftp> quit
221 Goodbye.
```

Se observa que tanto la carpeta local como la carpeta del servidor cuentan ahora con el archivo .txt. En este instante cerramos el servidor de Firezilla y procedemos a detener las capturas en el wireshark.

Ubicamos primero los 3 paquetes iniciales de inicio de sesion entre cliente y servidor:

No.	Time	Source	Destination	Protocol	Length	Info
34	40.755146	127.0.0.1	127.0.0.1	TCP	56	60920 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=1 SACK_PERM=1
35	40.755314	127.0.0.1	127.0.0.1	TCP	56	21 → 60920 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
36	40.755525	127.0.0.1	127.0.0.1	TCP	44	60920 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
37	40.757265	127.0.0.1	127.0.0.1	TLSv1.3	314	Application Data
38	40.757341	127.0.0.1	127.0.0.1	TCP	44	60919 → 14148 [ACK] Seq=525 Ack=1620 Win=2618112 Len=0
39	40.757547	127.0.0.1	127.0.0.1	FTP	121	Response: 220-FileZilla Server 1.8.2
40	40.757610	127.0.0.1	127.0.0.1	TCP	44	60920 → 21 [ACK] Seq=1 Ack=78 Win=8115 Len=0
41	40.795163	127.0.0.1	127.0.0.1	FTP	58	Request: OPTS UTF8 ON
42	40.795221	127.0.0.1	127.0.0.1	TCP	44	21 → 60920 [ACK] Seq=78 Ack=15 Win=2619648 Len=0
43	40.795559	127.0.0.1	127.0.0.1	TLSv1.3	251	Application Data
44	40.795592	127.0.0.1	127.0.0.1	TCP	44	60919 → 14148 [ACK] Seq=525 Ack=1827 Win=2617856 Len=0
45	40.795673	127.0.0.1	127.0.0.1	FTP	107	Response: 202 UTF8 mode is always enabled. No need to send this command
46	40.795702	127.0.0.1	127.0.0.1	TCP	44	60920 → 21 [ACK] Seq=15 Ack=141 Win=8052 Len=0
47	43.492657	127.0.0.1	127.0.0.1	FTP	55	Request: USER tomi
48	43.492757	127.0.0.1	127.0.0.1	TCP	44	21 → 60920 [ACK] Seq=141 Ack=26 Win=2619648 Len=0
49	43.493517	127.0.0.1	127.0.0.1	TLSv1.3	131	Application Data

Observamos los Flags del primer paquete

```
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
    [TCP Flags: ··········S·]
```

Se observa que el unico Flag que envia el cliente es el flag de SYN, que como se vio en la teoria, sirve para indicar la intecion de realizar un inicio de sesion. En cuanto al segundo paquete:

```
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
    [TCP Flags: ·····A··S·]
```

Se observa que el servidor devuelve un ACK, para confirmar la conexión, y tambien envia un flag de SYN, para tambien indicar que se quiere conectar con el cliente ()la conexión será con un flujo de informacion bidireccional. Por ultimo, el tercer paquete:

```
Acknowledgment number (raw): 3841208136
0101 .... = Header Length: 20 bytes (5)
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
    [TCP Flags: ·····A····]
```

Con lo cual el cliente devuelve un ACK y se establece una conexión exitosa entre el cliente y el servidor.

C) En la capa de transporte, además de la dirección IP, se debe identificar los servicios mediante el puerto. Los puertos son los identificadores de la capa de trasporte y cada aplicación tendrá asociado uno.
Recordemos que los puertos se pueden dividir por:

Nro puerto	Uso
1-1023	Bien conocidos(Asociados a una aplicación, por ejemplo Puerto 80:HTTP, 25:SMTP,FTP:21,DNS:53,TELNET:23,SSH:22)
1024-49151	Registrados (no están asociados por estándar a una aplicación, puedo asignarlos a aplicaciones propias)
49152-65535	Clientes privados, son los puertos que utilizan los clientes en una comunicación privada con un servidor.

En este caso, como se observa en la primer captura, el puerto utilizado para el servidor FTP es el 21 (el cual es bien conocido), y el utilizado por parte del cliente es el 60920.

D) Para seleccionar una aplicación, primero debemos tener en claro las diferencias y entre UDP y TCP y en que situaciones se utiliza cada uno. TCP es más confiable que UPD, ya que se fija en el orden de envio y llegada de los paquetes y tiene un ACK de validación con el cual por cada paquete se informa si el paquete llego, si llego duplicado, o si no llego. UDP solo manda los paquetes, sin asegurarse de que llegue. TCP hace validaciones, UDP no hace validaciones y por esta razon, UDP es más rápido que TCP, ya que no debe esperar dichas validaciones.
Dicho esto, una aplicación que utilice el protocolo UDP será aquella que priorice la velodad de envio de paquetes por soble la seguridad de su correcta recepcion. Puntualizando, un ejemplo puede ser whatsapp, especificamente, en su apartado de llamadas de voz, las cuales no necesita que se entregue el 100% de los paquetes (con que se puedan entender las personas que estan hablando es suficiente), pero si necesita que lo que se habla llegue practicamente al instante de un terminal a otro.