PRÁCTICA 3

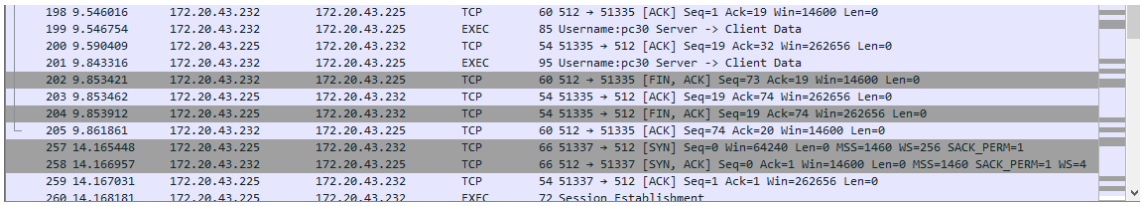
REDES

JAIME HERNÁNDEZ

EPS

Ejercicio 1: Norma RFC 1191

Utiliza el programa Rexec para ejecutar comandos UNIX (ls -a, pwd, who, etc.) en la máquina Linux 2 172.20.43.232. Utiliza el usuario "pcxx" y password "pcxx" (xx es el número del PC empleado por el alumno: 00, 01, 02, etc.)

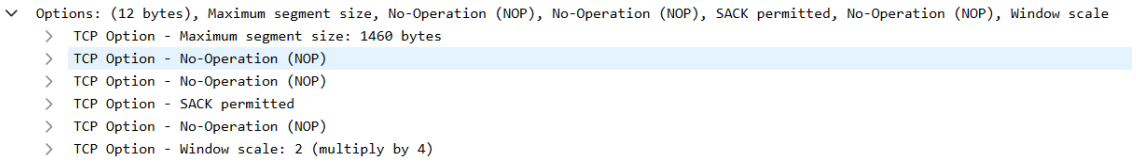


198	9.546016	172.20.43.232	172.20.43.225	TCP	60 512 → 51335 [ACK] Seq=1 Ack=19 Win=14600 Len=0
199	9.546754	172.20.43.232	172.20.43.225	EXEC	85 Username:pc30 Server -> Client Data
200	9.590409	172.20.43.225	172.20.43.232	TCP	54 51335 → 512 [ACK] Seq=19 Ack=32 Win=262656 Len=0
201	9.843316	172.20.43.232	172.20.43.225	EXEC	95 Username:pc30 Server -> Client Data
202	9.853421	172.20.43.232	172.20.43.225	TCP	60 512 → 51335 [FIN, ACK] Seq=73 Ack=19 Win=14600 Len=0
203	9.853462	172.20.43.225	172.20.43.232	TCP	54 51335 → 512 [ACK] Seq=19 Ack=74 Win=262656 Len=0
204	9.853912	172.20.43.225	172.20.43.232	TCP	54 51335 → 512 [FIN, ACK] Seq=19 Ack=74 Win=262656 Len=0
205	9.861861	172.20.43.232	172.20.43.225	TCP	60 512 → 51335 [ACK] Seq=74 Ack=20 Win=14600 Len=0
257	14.165448	172.20.43.225	172.20.43.232	TCP	66 51337 → 512 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
258	14.166957	172.20.43.232	172.20.43.225	TCP	66 512 → 51337 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=4
259	14.167031	172.20.43.225	172.20.43.232	TCP	54 51337 → 512 [ACK] Seq=1 Ack=1 Win=262656 Len=0
260	14.168181	172.20.43.225	172.20.43.232	EXEC	72 Session Establishment

Con el monitor de red, analiza y estudia la secuencia de paquetes TCP que se desencadenan. Comprueba que las secuencias de conexión y desconexión TCP que aparecen son las comentadas en el apartado 2.2 del documento de la práctica. Determina cuál es el valor de MSS que se negocia en los paquetes TCP SYN.

Como poemos observar en la captura de pantalla podemos ver el protocolo de la ip en la cual se han estado mando los datos de la IP que se nos han dado en este caso, Linux 2, ip.addr == 172.20.43.232, podemos observar tambien el envio y fin de los ACK que se han ido mandando.

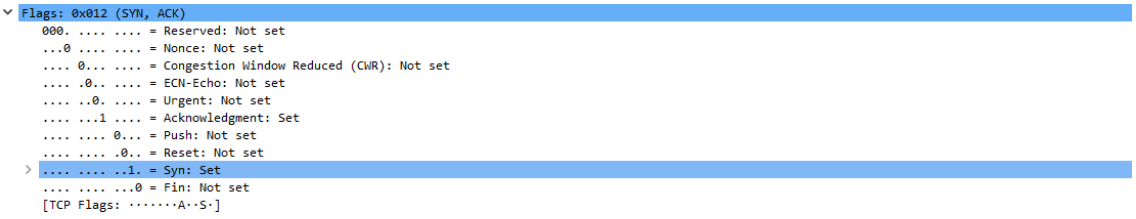
Comprueba también los valores de puertos utilizados (número asignado de puerto cliente y número de puerto utilizado por el servidor), como varían los números de secuencia de byte y de ACK, y los flags activados en las cabeceras TCP.



```
Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
    TCP Option - No-Operation (NOP)
    TCP Option - No-Operation (NOP)
    TCP Option - SACK permitted
    TCP Option - No-Operation (NOP)
    TCP Option - Window scale: 2 (multiply by 4)
```

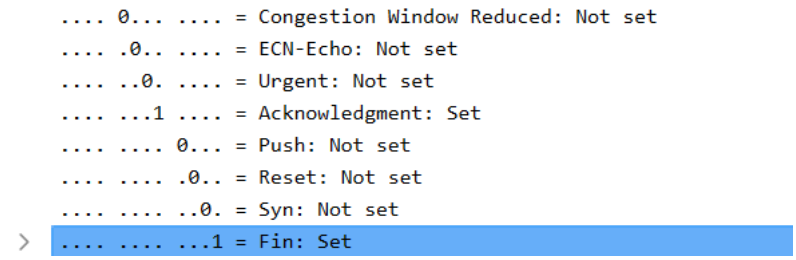
El MSS es de 1460 bytes, en este caso ambas máquinas tenían el mismo MTU (1500).

El servidor utiliza el puerto especificado para el protocolo UTC en la herramienta rexec, la máquina origen utiliza el puerto. Los flags varían dependiendo del propósito del paquete envíado. En la segunda captura tambien podemos obsevar los flags qe se activan, de manera más ampliada se observa en la siguiente imagen:



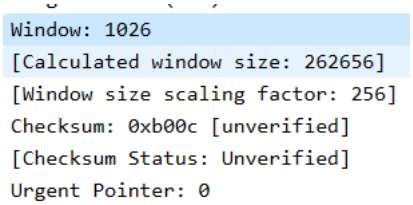
```
Flags: 0x012 (SYN, ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ·····A··S·]
```

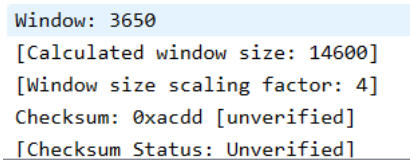
Aquellos destinados a cerrar las conexiones tienen el bit Fin a 1, en la siguiente captura se puede ver que más de una flag puede ser activada:



Para recibir correctamente los paquetes la ventana debe ser mayor a la del servidor para, en caso de error, poder almacenar en el buffer los paquetes que se siguen recibiendo hasta recibir de nuevo el paquete que causó error en un primer lugar.

Analiza los valores de las ventanas de receptor anunciadas. ¿Son iguales en el cliente y en el servidor? ¿Cuál es más grande y por qué piensas que son diferentes?





La información recibida es de varios miles de bytes y se enviará en paquetes TCP de gran tamaño. ¿ Fragmenta el protocolo IP estos paquetes TCP grandes ? ¿ Por qué no ?

El protocolo IP no es responsable de la fragmentación de los paquetes TCP, los paquetes TCP tienen el bit DF en el encabezado IP establecido en 1. La fragmentación de estos paquetes depende del MSS. El MSS se calcula restando el peso de los encabezados IP y TCP del segmento físico con la MTU más pequeña. Si los segmentos físicos intermedios deben fragmentarse porque los encabezados IP de estos paquetes indican que no deben fragmentarse, se debe devolver un error de fragmentación ICMP a la máquina de origen y la máquina de origen debe fragmentarlo. Todos los paquetes enviados a partir de ahora deben ser paquetes fragmentados para que sus paquetes puedan enrutarse sin dividirlos en segmentos intermedios.

Norma rfc

```
C:\Users\EPS\Desktop>ftp 172.20.41.243
Conectado a 172.20.41.243.
220 bftpd 4.2 at 172.20.41.243 ready.
503 USER expected.
Usuario (172.20.41.243:(none)): pc25
331 Password please.
Contraseña:
230 User logged in.
ftp> rename rfc1191.txt old.txt
350 File exists, ready for destination name
250 OK
ftp> put rfc1191.txt
200 PORT 172.20.43.222:56687 OK
150 BINARY data connection established.
226 File transmission successful.
ftp: 11453 bytes enviados en 22.27segundos 0.51a KB/s.
ftp> quit
221 See you later...
```

Determina con el monitor de red qué valor de MSS se ha negociado en la conexión TCP

```
Options: (12 bytes), Maximum segment size, No-Op
  TCP Option - Maximum segment size: 1460 bytes
  TCP Option - No-Operation (NOP)
  TCP Option - Window scale: 0 (multiply by 1)
  TCP Option - No-Operation (NOP)
  TCP Option - No-Operation (NOP)
  TCP Option - SACK permitted
```

El MSS acordado ha sido de 1460 bytes.

¿Aparecen paquetes ICMP fragmentation needed and the bit don't fragment was set? ¿Quién envía el mensaje ICMP de error?

El Router 1 envía varios errores ICMP Fragmentation needed, por lo tanto el MTU de R1 será menor que el MTU de las máquinas que establecen la conexión.

¿Cómo afecta este mensaje ICMP al tamaño de los paquetes TCP intercambiados entre tu PC y el equipo Linux 3 (172.20.41.243)?

Desde ese momento los paquetes TCP tienen un MSS de (MTU - Cab. TCP - Cab. Ip)

600 – 40 = 560

```
MTU of next hop: 600
Internet Protocol Version 4, Src: 172.20.43.199,
Transmission Control Protocol, Src Port: 49251,
```

¿Reenvía tu PC algún paquete TCP al equipo Linux 3?

Después de conocer el nuevo MSS, mi PC reenvía los paquetes al equipo Linux 3. En este caso intenta envíar cada paquete con su MSS inicial y luego, al recibir el error ICMP, lo reenvía con el mismo valor de secuencia con el MSS adecuado. La fragmentación no puede ser realizada por el protocolo Ip pues los paquetes del protocolo TCP tienen el bit DF activado.

```
1174 49251 → 33017 [ACK] Seq=4062643428 Ack=2918825536 Win=131328 Len=1120
  70 Destination unreachable (Fragmentation needed)
 614 49251 → 33017 [ACK] Seq=4062643428 Ack=2918825536 Win=131328 Len=560
  60 33017 → 49251 [ACK] Seq=2918825536 Ack=4062643988 Win=33640 Len=0
1174 49251 → 33017 [ACK] Seq=4062643988 Ack=2918825536 Win=131328 Len=1120
  70 Destination unreachable (Fragmentation needed)
 614 49251 → 33017 [ACK] Seq=4062643988 Ack=2918825536 Win=131328 Len=560
```

¿Fragmenta IP algún paquete TCP?

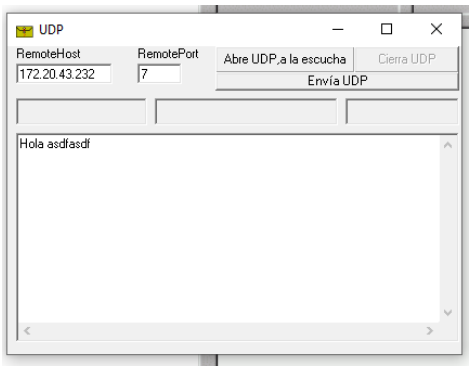
En ningún caso el protocolo Ip fragmenta paquetes TCP, esto sería ineficiente y es por eso que se utilizan los paquetes ICMP de error. Además, los paquetes enviados tienen el bit DF a 1 por norma del protocolo.

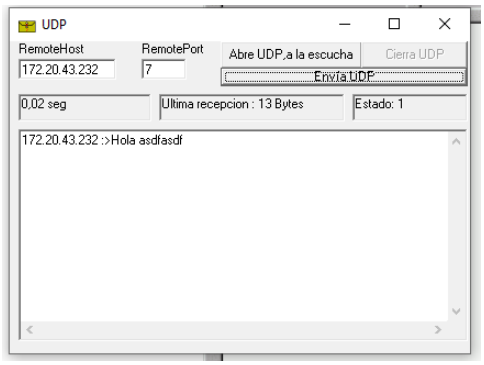
Intentar acceder al servidor Web de la máquina 172.20.43.232 empleando un navegador web (Firefox, Intenet Explorer, etc.) con la direcció http://172.20.43.232/. Determinar qué secuencia de paquetes se intercambian en la conexión TCP y por qué.

EJERCICIO 2: PROTOCOLO UDP

Con el monitor de red analizar la secuencia de paquetes UDP que se desencadenan cuando se envía como datos un texto de menos de 100 caracteres. Comparar los valores de los campos de tamaño de las cabeceras IP y UDP.

En el protocolo UDP no se utilizan paquetes de sincronización a diferencia de en el protocolo TCP. Filtrando por los paquetes eco únicamente se ve un Request desde la máquina y el correspondiente response de la máquina destino.



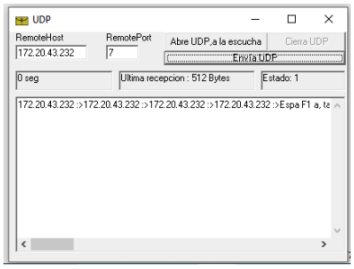


La cabecera IP tiene una longitud de 20 bytes y contiene información esencial para el enrutamiento y la entrega de los datos. Incluye detalles como la dirección IP de origen, la dirección IP de destino, la longitud total del paquete, y otros campos necesarios para la gestión de la red.

Por otro lado, la cabecera UDP es más ligera, ocupando solo 8 bytes. Esta cabecera se centra en proporcionar la información necesaria para el protocolo de transporte UDP. Incluye los puertos de origen y destino, la longitud del paquete y un campo de suma de verificación que se utiliza para garantizar la integridad de los datos transmitidos.

En resumen, mientras que la cabecera IP se enfoca en aspectos más generales de la entrega de datos a través de la red, la cabecera UDP se centra en la información específica necesaria para la comunicación punto a punto, manteniendo una estructura más compacta.

Realizar el mismo procedimiento que en el apartado anterior pero enviando un texto mucho más grande (sobre 2000 bytes) al puerto UDP 7. Para ello puede copiarse parte de algún fichero de texto en el panel inferior de la aplicación Udp ¿Se produce fragmentación IP de los paquetes UDP ?. Analiza el valor del campo longitud de la cabecera UDP y del campo longitud total de la cabecera IP en los paquetes enviados y recibidos.



A diferencia del protocolo TCP, el protocolo UDP no asume la responsabilidad de dividir en fragmentos los paquetes que exceden el Tamaño Máximo de Unidad de Transmisión (MTU). En lugar de eso, es el propio protocolo IP el encargado de llevar a cabo la fragmentación de dichos paquetes en caso de que superen el límite establecido. Esta diferencia destaca el enfoque más simplificado de UDP, ya que delega la tarea de fragmentación al nivel de red.

Realizar un nuevo envío de un texto corto al puerto UDP 7, pero dirigido a la dirección de broadcast de la red local. ¿ Qué estaciones contestan a este envío ?

A diferencia de otros protocolos, como TCP, el protocolo UDP tiene la capacidad de enviar paquetes a la dirección de difusión, permitiendo que todos los dispositivos en la red los reciban. No obstante, para prevenir posibles ataques que podrían explotar esta característica como una vulnerabilidad, el protocolo UDP implementa una medida de seguridad: los equipos no responderán a los mensajes UDP dirigidos a la dirección de difusión. En este escenario, solo un equipo específico proporcionará una respuesta, evitando así posibles abusos de la función de difusión.

Intenta enviar un texto al puerto 80 de la máquina Linux 2 (172.20.43.232) empleando el programa Udp. Determina la secuencia de paquetes que se intercambian entre tu equipo y el equipo 172.20.43.232.

Cuando el equipo con la dirección IP 172.20.43.232 recibe un intento de envío de un paquete UDP al puerto http de la máquina Linux 2, el protocolo responde con un mensaje ICMP de error "Port Unreachable". Esta respuesta indica que el puerto específico al que se intenta acceder no está disponible en la máquina de destino. A diferencia del protocolo TCP, en el caso del protocolo UDP, la notificación de error no se recibe a través del mismo protocolo, sino que se apoya en el protocolo ICMP para comunicar la inaccesibilidad del puerto.

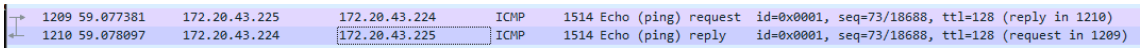
EJERCICIO 3: VELOCIDAD DE TRANSMISIÓN

Determina de forma experimental la velocidad de transmisión en la red Ethernet del aula L24 mediante el envío de tramas de un tamaño determinado con la aplicación ping. Para ello emplea el comando: ping -n 1 -l 1472 direccion_IP_PC_AulaL24 Donde dirección_IP_PC_AulaL24 es la dirección IP de algún PC del aula del laboratorio. La red Ethernet del aula L24 emplea Ethernet 100BaseTX (no así los routers del laboratorio que emplean Ethernet 10BaseT).

```
C:\>ping -n 1 -l 1472 172.20.43.224

Haciendo ping a 172.20.43.224 con 1472 bytes de datos:
Respuesta desde 172.20.43.224: bytes=1472 tiempo<1m TTL=128

Estadísticas de ping para 172.20.43.224:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```



Para conocer cuántos bits se han enviado es necesario saber cuántos bytes ocupa el mensaje ICMP Echo Request enviado, en este caso se sabe gracias a WireShark que las cabeceras y datos ocupan 1514 bytes, a los que hay que sumar los 8 de preámbulo y los 4 de CRC. Esto deja un total de 1526 bytes multiplicado por 2 ya que se necesita el tamaño de petición y respuesta: 3052 bytes. Como se pide en bits 24416 bits.

Para poder saber la velocidad de respuesta tenemos que restar el Reply del Request por lo que el resultado nos daría 0,000716 s.

Vt = 24416 bits / 0. 000716 s = 34,14 Mbps