REDES – Práctica 8

Grado en Ingeniería Informática

Contenido

1. El analizador de protocolos WireShark

Los analizadores de protocolos o de red, también conocidos como “sniffers” son herramientas de gran ayuda para los administradores de las redes de computadores, ya que permiten el análisis detallado de muchos factores del comportamiento de las mismas. Estas aplicaciones permiten capturar una copia de los paquetes que circulan por la red para su análisis posterior. Muchos de ellos incluyen una interfaz gráfica capaz de mostrar los campos de los protocolos de comunicación de los distintos niveles, obtener estadísticas de utilización y facilitar considerablemente el posterior análisis de los datos capturados. De este modo se facilita la detección de problemas, así como la depuración del software de red durante su fase de elaboración. Por ejemplo, un administrador de red que detecte que las prestaciones de la red son bajas puede utilizar uno de estos analizadores para detectar qué segmentos de la red, protocolos y máquinas están generando más tráfico, y de esa forma llevar a cabo las acciones necesarias, o bien verificar el correcto funcionamiento de los diferentes dispositivos de red (*hosts*, servidores, *routers*, cortafuegos, etc).

Desde el punto de vista docente, los analizadores de protocolos permiten ver de forma práctica los protocolos de comunicación ya presentados en las clases de teoría, así como las relaciones entre los protocolos de distinto nivel. En esta práctica se pretende adquirir las capacidades necesarias para capturar paquetes usando la herramienta *WireShark* y conseguir en primer lugar que se conozcan las acciones (secuencias de intercambio de paquetes) generadas por la ejecución de las órdenes de red más frecuentes. En segundo lugar, dado que por la red viajan multitud de paquetes, será necesario seleccionar aquellos que nos resulten de interés. Por ello vamos a aprender a capturar paquetes utilizando los filtros que nos proporciona *WireShark*, de manera que aceptaremos unos paquetes y desecharemos otros. En tercer lugar, se pretende introducir al alumno en la interpretación del contenido de los paquetes capturados para afianzar los conceptos relativos a los protocolos estudiados en clase. Todo ello permitirá poner en práctica los conocimientos de teoría, adquiriendo una mayor comprensión de los procesos que ocurren en la red cuando se llevan a cabo diversas acciones a nivel de usuario.

1.1 El entorno de WireShark

A la hora de elegir un analizador de protocolos nos encontramos con una abundante oferta, tanto de productos comerciales, como de software de libre distribución. Uno de los más populares, y el elegido para la práctica es *WireShark.* Se trata de un producto gratuito y muy

versátil, que puede descargarse desde http://www.wireshark.org. Está disponible tanto para sistemas Windows como Unix, como macOS, y permite no sólo capturar tráfico de una red, sino también filtrarlo y analizarlo. Además, permite leer ficheros de datos recogidos con otros analizadores de protocolos como *tcpdump*.

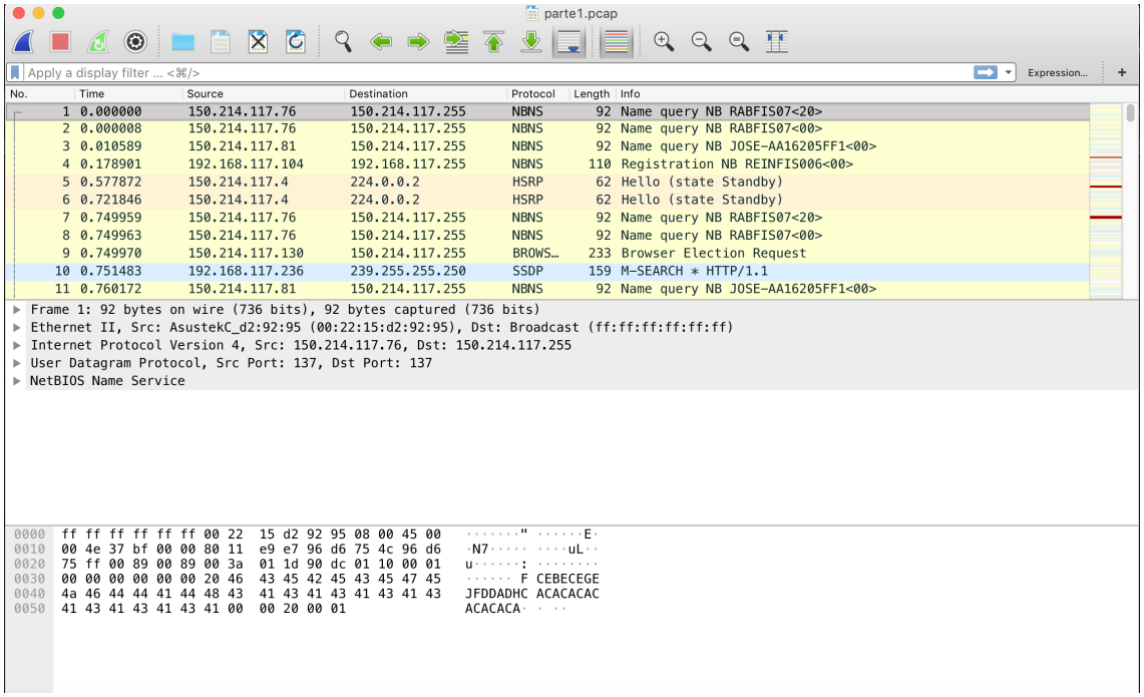


Figura 1. Ventana principal del programa

Comenzaremos comentando el aspecto habitual del programa. WireShark comprende tres ventanas o áreas principales.

1) ***La ventana superior es la lista de los paquetes.*** Muestra una breve descripción de cada paquete capturado. Pulsando en alguno de los paquetes de esta lista podemos controlar lo que se visualiza en las dos ventanas restantes.
2) ***La ventana intermedia muestra con mayor detalle el paquete seleccionado en la primera ventana***. Indica los protocolos empleados en los distintos niveles de la arquitectura, así como los valores de cada uno de los campos de cada protocolo.
3) Por último, ***la ventana inferior muestra el valor de los datos, en hexadecimal y en ASCII, del paquete seleccionado en la ventana superior***, y marca en negro los datos seleccionados en la ventana intermedia.

Además de estas tres ventanas, WireShark ofrece un filtrado en pantalla de los paquetes capturados en función del tipo de paquetes y/o contenido de sus campos. Este filtrado *a posteriori* es complementario al filtrado de paquetes en el momento de la captura que se verá en la siguiente sección.

1.2 Cómo capturar paquetes

Para realizar una captura de paquetes hay que acceder al menú *Capture* y allí seleccionar la opción *Option*. En esta ventana podemos especificar, si fuera necesario, diversos parámetros relacionados con la captura.

El primer parámetro que podemos especificar es el interfaz. Aquí se especificará la tarjeta de red sobre la que queremos realizar la captura. Esta opción sólo tiene sentido si disponemos de varias tarjetas de red (posiblemente conectadas a diferentes redes).

Otra opción que podemos detallar es si deseamos realizar una captura en modo promiscuo o no. En el modo promiscuo el programa capturará cualquier paquete que sea visible a la tarjeta de red, independientemente de si está destinado a ella o no. Por el contrario, podemos seleccionar la captura de únicamente aquellos paquetes que van destinados a, o que provienen de, nuestra tarjeta de red. La captura de otros paquetes con un origen o destino diferente al nuestro depende del dispositivo intermediario que conecte las computadoras (hub, switches, routers) y de los privilegios que se posean

En esta ventana también podemos introducir un filtro de captura, con el fin de procesar después más fácilmente la información obtenida. El uso de este tipo de filtros lo describiremos más adelante.

Otra posibilidad que nos ofrece esta ventana es la de volcar los paquetes capturados a un fichero. Esto puede resultar interesante para guardar un registro del tráfico capturado. No obstante, también se pueden almacenar las capturas después de realizarlas, desde el menú *File/Save*. Otras opciones que esta ventana nos permite detallar están relacionadas con la visualización en pantalla de la captura. Podemos optar por ver en tiempo real los paquetes que se van capturando y también podemos elegir que se realice un desplazamiento vertical automático de la pantalla (*scrolling*).

Finalmente, con el fin de terminar la captura, podemos indicar que ésta termine automáticamente cuando se hayan capturado cierto número de paquetes, o se haya capturado una cantidad determinada de Kbytes, o bien cuando haya transcurrido cierta cantidad de tiempo. En caso de no seleccionar ninguna de las opciones, tendremos que finalizar la captura de forma manual.

Ejercicio 1

Abre el fichero parte1.pcap. Observa los tipos de paquetes que se tiene, centra la atención en el primer paquete recibido y consultando la cabecera relativa al protocolo IP, rellena los siguientes datos del mismo:

Primer paquete IP recibido	
Dirección IP origen	
Dirección IP destino	
Protocolo	
Tamaño cabecera	
Tamaño total	
TTL	
Identificador	

1.3 Filtros de captura y de pantalla

Al intentar analizar el tráfico de cualquier red, resulta habitual encontrarse gran cantidad de paquetes que emplean protocolos en los que no estamos interesados. Tal cantidad de tráfico dificulta el análisis de los paquetes capturados y aumenta innecesariamente el tamaño de los ficheros de captura, por lo que se hace indispensable filtrar toda esa información.

Wireshark ofrece numerosas posibilidades de filtrado de información, que básicamente consisten en la selección de protocolos, la definición de un filtro de captura y la definición de un filtro de presentación de la información.

Protocolos

Como primer nivel de filtrado, podemos escoger los protocolos con los que deseamos trabajar. La lista completa de protocolos que maneja Wireshark puede verse en un cuadro de diálogo al

que podremos acceder mediante la opción de menú Analyze → Enabled Protocols… En dicho cuadro de diálogo, se puede activar o desactivar la utilización de los protocolos que deseemos.

A la hora de activar o desactivar protocolos, debemos tener en cuenta la advertencia que aparece en este cuadro de diálogo, y que indica que, si desactivamos un protocolo, no aparecerán los protocolos de los niveles superiores que dependan de él. Por ejemplo, si desactivamos el protocolo de nivel de transporte, TCP, no aparecerán tampoco todos los protocolos de nivel de aplicación que dependan de él, como HTTP y SMTP.

Captura

El siguiente nivel de filtrado que ofrece Wireshark se aplica al proceso de captura de los paquetes de red. Podemos definir un filtro que capture únicamente los paquetes de un determinado protocolo o destinados a un determinado ordenador o puerto. La utilización de un filtro se realiza en el cuadro de diálogo de opciones de captura, introduciendo la expresión del filtro en el cuadro de texto situado al lado del botón **Capture Filter.**

En la tabla que se muestra a continuación se indican alguna de las opciones disponibles para crear filtros de captura.

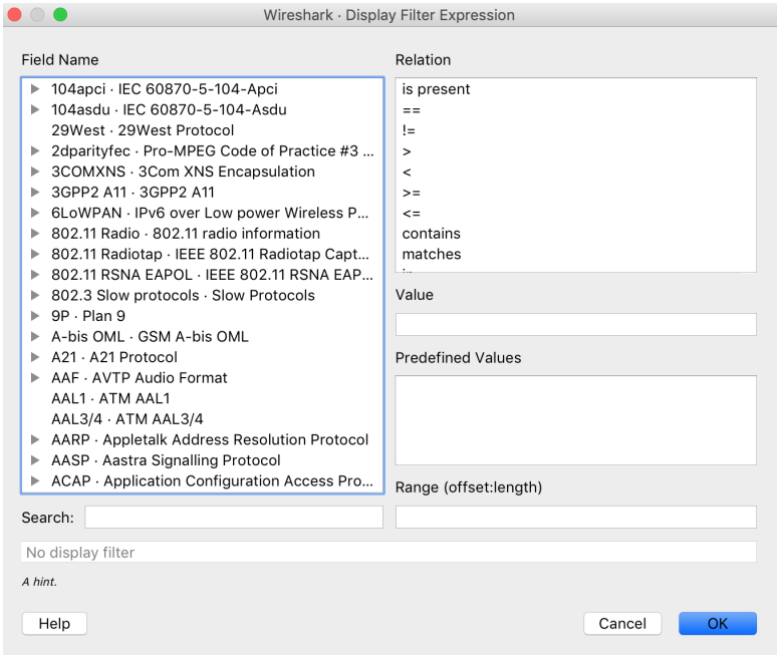
EXPRESIÓN	DESCRIPCIÓN
[src|dst] host <host>	• Filtrar el tráfico desde o hacia el host especificado por <host>. • Opcionalmente, la expresión host puede ir precedida de src o dst, para indicar, respectivamente, que sólo se está interesado en el tráfico desde el host (source) o hacia el host (destination).
ether [src|dst] host <ehost>	• Filtra el tráfico desde o hacia una dirección Ethernet. La opción src|dst tiene el significado contado anteriormente.
[tcp|udp] [src|dst] port <port>	• Filtra el tráfico en el puerto (port) especificado. • El archivo /etc/services contiene la lista con puertos que corresponde a cada aplicación. • La opción tcp|udp permite elegir el tráfico TCP o UDP respectivamente (si se omite, se seleccionarán todos los paquetes de ambos protocolos). • La opción src|dst tiene el significado contado anteriormente
ip|ether proto <protocolo>	• Filtra el protocolo especificado (IP o Ethernet)

FILTRO	DESCRIPCIÓN
host 172.16.222.10	Captura tráfico desde y hacia el host 172.16.222.10
dst host 172.18.5.4	Captura tráfico sólo hacia el host 172.18.5.4
port 21	Captura tráfico FTP
arp	Captura tráfico ARP
port 22 and host 10.0.0.5	Captura tráfico ssh desde o hacia el host 10.0.0.5

Presentación

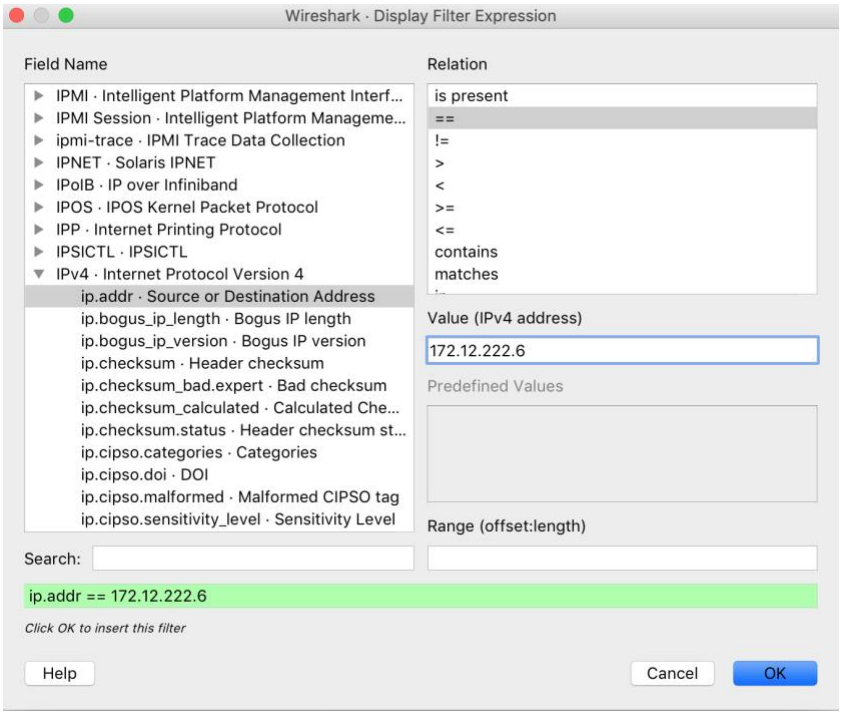
El último nivel de filtrado del que disponemos en Wireshark es el de presentación de los paquetes. Podemos definir un filtro mediante el cual seleccionemos, para que se vean en el panel principal, únicamente aquellos paquetes de datos que nos interesa analizar. Este tipo de filtro es el más completo y en su expresión se pueden utilizar la mayor parte de los parámetros de los protocolos que estamos analizando.

Para utilizar un filtro de presentación podemos escribir su expresión directamente en el recuadro de texto del filtro y aplicarlo mediante el botón Apply. Tenemos otra alternativa, que consiste en pulsar el botón de filtro denominado **<<*expression*>>.** Al hacerlo aparece un cuadro de diálogo de definición de filtro, como el que se muestra.



Si no conocemos la sintaxis de los filtros o los parámetros que podemos emplear en los mismos, este cuadro de diálogo nos permite crear una expresión de forma visual.

Finalmente, también se pueden definir diversos filtros, asignándoles distintos nombres para su posterior aplicación, mediante la opción de menú **Analyze → Display Filters Expression…**



En la figura vemos cómo definir un filtro para mostrar los paquetes de datos cuya dirección IP de origen o de destino sea la 172.12.222.6. La expresión que se genera para este filtro es:

ip.addr == 172.12.222.6

Algunos ejemplos de filtros de presentación se muestran a continuación:

FILTRO	DESCRIPCIÓN
tcp	Solamente presentar los paquetes de datos del protocolo TCP.
ip.proto == 1 and ip.src_host == 172.16.222.1	Paquetes cuyo protocolo IP sea el 1 (el correspondiente a ICMP) y cuya dirección IP de origen sea la 172.16.222.1
(ip.addr == 172.16.222.1 || ip.addr == 172.16.222.2) and (tcp.port == 1028 || tcp.port == 80).	Paquetes cuya dirección IP de origen sea la 172.16.222.1 y su dirección IP de destino sea la 172.16.222.2 (o viceversa) y cuyo puerto TCP de origen sea el 1028 y su puerto TCP de destino sea el 80 (o viceversa).

En los ejercicios de esta sección se proponen el uso de diferentes filtros. Para acceder a una documentación detallada acerca de los filtros y sus posibilidades, se puede consultar el manual en línea con la orden tcpdump, (***man tcpdump)***.

Ejercicio 2

Plantea un filtro en la captura para que solamente acepte aquellos paquetes que lleven estrictamente como dirección origen o destino la dirección IP de vuestra máquina.

Respuesta	

Ejercicio 3

Crea un filtro de captura para que solamente capture las consultas realizadas a los DNS (paquetes con origen o destino el puerto 53 UDP).

Respuesta	

Ejercicio 4

Crea un filtro de captura que solamente capture las consultas realizadas a los DNS (paquetes con origen o destino el puerto 53 UDP) y cuyo origen de la comunicación sea tu máquina.

Respuesta	

Ejercicio 5

Crea un filtro que solamente capture paquetes UDP.

Respuesta	

2. Trabajando con diferentes protocolos

En este apartado vamos a emplear *WireShark* para observar el funcionamiento de algunos de los protocolos vistos en la práctica anterior.

Para entender lo que está ocurriendo en cada captura es conveniente preguntarse qué máquinas están involucradas en la captura (sus direcciones IP), qué puertos se están utilizando, qué protocolo del nivel de transporte está contenido en el datagrama (si lo hay) y qué aplicación ha generado dicho datagrama, etc.

2.1 Protocolo ICMP

El Protocolo de Mensajes de Control y Error de Internet, ICMP, se utiliza para controlar si un paquete no puede alcanzar su destino, si su vida ha expirado, si el encabezamiento lleva un valor no permitido, si es un paquete de eco o respuesta, etc. Es decir, se usa para manejar mensajes de error y de control necesarios para los sistemas de la red, informando con ellos a la fuente original para que evite o corrija el problema detectado.

La orden **ping** genera paquetes ICMP de tipo *echo request* y *echo reply*. A continuación, comprobaremos el funcionamiento de la misma analizando los paquetes que genera.

Ejercicio 1

Abre el archivo parte1.pcap, en la captura mostrada se encuentra el envío producido por el comando **ping -c 4 209.85.229.104**.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación para que sólo aparezcan los paquetes enviados a la dirección 209.85.229.104. Para ello, bastará con escribir (ip.addr == 209.85.229.104) en el campo de texto de filtro de la ventana principal y pulsar el botón Apply.

¿Cuántos mensajes ICMP se producen, prestando especial atención a los campos **tipo**, **código**, y **bytes de datos?** Asimismo, analiza las cabeceras IP de cada uno de ellos, y en concreto los campos **longitud de la cabecera** y **longitud total**.

Respuesta	

Ejercicio 2

Para comprobar el funcionamiento del protocolo en otras condiciones, en el mismo fichero que tenemos abierto se encuentra la captura relativa a la ejecución del comando **ping -l (tamaño-*paquete*) –c (*número-de-paquetes*) 192.168.117.205**. Para controlar el número de mensajes que se envían sería conveniente que en el filtro de presentación se indicase (ip.addr == 192.168.117.205).

¿Cuántos mensajes ICMP se producen ahora en cada envío y recepción?, ¿cuántos paquetes se han enviado?, Analiza los parámetros de la cabecera que te indican cómo están fragmentados los mensajes, comprueba para ello los campos **don't fragment, more fragment y fragment offset** de un envío y de una recepción.

Respuesta	

2.2 Protocolo DNS

El protocolo DNS, que emplea el puerto 53 de UDP, se emplea para poder denominar a los computadores mediante nombres simbólicos, sin tener que recordar las direcciones IP correspondientes a cada computador. A continuación, observaremos el funcionamiento del DNS**.**

Ejercicio 3

Continuando con el fichero parte1.pcap, vamos a analizar la información que se genera en la captura cuando se ejecuta el comando **ping -c 1 www.google.es**.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación para que sólo aparezcan los paquetes de este protocolo (***dns***).

¿Cuántos mensajes DNS se generan? ¿Qué tipo de consulta se realiza? El tipo de las consultas viene detallado en el campo ***Queries*** de la cabecera del protocolo DNS. ¿En qué puerto y dirección IP está localizado el servidor de nombres? ¿Qué dirección IP tiene el dominio www.google.es? ¿Qué protocolo de capa de transporte utiliza los paquetes DNS?

Respuesta	

2.3 Protocolo ARP

Para que un datagrama llegue a su destino es necesario especificar, además de la dirección IP destino, la dirección física del adaptador de red que debe recibir la trama en la que viaja el datagrama. Este adaptador de red puede ser el del *host* destino o bien el de un *router* intermedio. Precisamente, para averiguar la dirección física que corresponde a una dirección IP determinada se creó el protocolo ARP, como se vio en la práctica anterior.

Ejercicio 4

En el fichero parte1.pcap, se encuentra almacenado los paquetes producidos por la ejecución del comando **ping –c 3 192.168.117.205**.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación para que sólo aparezcan los paquetes del protocolo ARP (***arp***).

¿Se han generado paquetes ARP para resolver la dirección IP indicada?

Respuesta	

Analice el mensaje ARP que pregunta por la dirección MAC de la dirección IP indicada en el destino del comando ping y obtenga la siguiente información de los campos de dicho protocolo. ¿Cuál es la dirección MAC del remitente?

Respuesta	

Revisa de nuevo los campos del datagrama de dicho protocolo y determina si se trata de un mensaje de petición o de una respuesta ARP. ¿Qué campo nos da esta información?

Respuesta	

Busca el paquete ARP que es la respuesta de la petición realizada y obtén cuál es la dirección física y dirección IP del destinatario del mensaje, ¿a quién corresponde dicha dirección física?

Respuesta	

2.4 Protocolo UDP

Es el protocolo no confiable empleado en la capa de transporte de TCP/IP.

Ejercicio 5

Para analizar la estructura de este protocolo con Wireshark vamos a abrir el archivo udp.pcap donde se encuentra almacenado la captura de una comunicación UDP.

Analice los paquetes y comente cuáles son las direcciones IP y puertos involucrados en la comunicación.

Respuesta	

¿Cuál es el número de paquetes UDP y el número de bytes de datos intercambiados?, ¿qué datos se han mandado?

Respuesta	

2.5 Protocolo TCP

Es el protocolo confiable empleado en la capa de transporte de TCP/IP.

Ejercicio 6

Para analizar la estructura de este protocolo con Wireshark vamos a abrir el archivo tcp.pcap donde se encuentra almacenado la captura de una comunicación TCP.

Analice los paquetes y responda, suponiendo que se emplea la comunicación cliente/servidor ¿Cuál es la dirección IP y el puerto del cliente TCP?, y ¿la dirección IP y el puerto del servidor TCP?

Respuesta	

¿Cuántos segmentos TCP se han enviado desde el cliente al servidor? y ¿desde el servidor al cliente?

Respuesta	

¿Qué extremo cierra primero la conexión?

Respuesta	

En el menú de Wireshark, seleccionando *Edit* → *Preferences* → *Protocols* → *TCP,* puedes desactivar la opción *Relative Sequence Numbers*. De esta forma podrás observar los números de secuencia reales, en lugar de los números relativos que muestra por omisión Wireshark.

¿Cuántos bytes de datos envía el cliente al servidor? Indica cuáles son los números de secuencia del SYN y del FIN que envía el cliente y qué relación tienen con la cantidad de datos enviada al servidor.

Respuesta	

¿Cuántos bytes de datos envía el servidor al cliente? Indica cuáles son los números de secuencia del SYN y del FIN que envía el servidor y qué relación tienen con la cantidad de datos enviada al servidor.

Respuesta	

Cuando hayas observado los números de secuencia reales, vuelve a activar la opción *Relative Sequence Numbers* y vuelve a contestar las últimas dos preguntas realizadas.

Respuesta	

2.6 Protocolo HTTP

Es el protocolo empleado para acceder a páginas web a través de Internet. Estudiaremos el funcionamiento general de este protocolo sin profundizar demasiado en los detalles concretos.

Ejercicio 7

Para analizar la estructura de este protocolo con Wireshark vamos a abrir el archivo parte2.pcap donde se encuentra almacenado la captura del acceso a la web localizada en 195.81.202.109.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación para que sólo aparezcan los paquetes de este protocolo.

Wireshark ofrece una funcionalidad que facilita la creación de la expresión del filtro si queremos que aparezcan únicamente los datos de una sesión TCP (concretamente, el acceso a una página web). Para utilizar esta funcionalidad debemos identificar algún paquete perteneciente a la sesión que nos ocupa, fácilmente localizable porque la dirección IP de destino debe ser 195.81.202.109 y el puerto TCP de destino el 80 (el habitual en la mayor parte de los servidores web de Internet).

Una vez localizado algún paquete de la sesión, bastará con pulsar con el botón derecho del ratón sobre él en el panel y seleccionar la opción follow TCP Stream en el menú emergente que aparecerá. Al hacerlo, además de crearse la expresión de filtro adecuada en el recuadro de texto del filtro, aparecerá una ventana adicional en la que se mostrará todo el contenido de la sesión TCP correspondiente. En esta ventana podemos seleccionar ver toda la conversación que ha tenido lugar entre nuestro navegador web (cuyas peticiones se muestran en color rojo) y el servidor web (cuyas respuestas se muestran en color azul). Si cerramos esta ventana y volvemos a la página principal se pueden ver todos los mensajes que se han llevado a cabo.

Identifica el primer mensaje http (después de seleccionar la opción ***follow TCP Stream***) y obtén de este mensaje la dirección IP del destino, la versión del protocolo HTTP que se está utilizando y la dirección IP del origen.

Respuesta	

Identifica un mensaje cuyo método que solicita un archivo del servidor, estos mensajes se identifican por llevar el comando GET. Consulta el campo URI del protocolo http y especifica la dirección donde se localiza dicho archivo.

Respuesta	

2.7 Protocolo TELNET

El propósito del protocolo TELNET es permitir un método estándar de comunicar entre sí terminales y procesos orientados a terminal. Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajan por la red como texto plano (cadenas de texto sin cifrar). Esto facilita que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas.

Ejercicio 8

Para analizar la estructura de este protocolo con Wireshark vamos a abrir el fichero telnet.pcap.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación indicando "telnet" para mostrar únicamente mensajes de este protocolo.

Mira la información de los paquetes que has enviado con este protocolo y determina el puerto que se utiliza para telnet.

Respuesta	

Lee el contenido de los paquetes, para ello tendrás que mirar el contenido en ASCII del campo ***Data*** del protocolo telnet. Localiza la solicitud de login y password del servidor, así como la lectura de los datos que se enviaron, ten presente que puedes necesitar varios paquetes para obtener finalmente el login y password que se escribieron. ¿Qué nombre de usuario y qué contraseña utilizó para iniciar sesión?

Respuesta	

2.8 Protocolo SSH

SSH permite a los usuarios registrarse en sistemas de host remotamente. A diferencia de telnet, visto anteriormente, SSH encripta la sesión de registro imposibilitando que alguien pueda obtener una contraseña de texto. El uso de métodos seguros para registrarse remotamente a otros sistemas hace disminuir los riesgos de seguridad para ambos sistemas y el sistema remoto.

Ejercicio 9

Para analizar la estructura de este protocolo con Wireshark vamos a seguir los mismos pasos que para el protocolo anterior, pero en este caso abriremos el archivo ssh.pcap.

Para facilitar el análisis, es recomendable aplicar un filtro de presentación para que sólo aparezcan los paquetes de este protocolo (***ssh***).

Mira la información de los paquetes que has enviado con este protocolo y determina el puerto que se utiliza para ssh.

Respuesta	

Lee el contenido de los paquetes, para ello tendrás que mirar la ventana que muestra el contenido ASCII de los campos del protocolo ssh. ¿Encuentras el login y las claves que se han introducido para iniciar la sesión remota?

Respuesta	