Universidad de Alcalá

Departamento de Automática

ARQUITECTURA DE REDES
Laboratorio

Práctica 3: Práctica con WireShark

Grado en Ingeniería Informática
Curso 2015/16

OBJETIVOS

1. Aprender a utilizar el analizador de trafico WireShark

CONTENIDOS

- Analizadores de protocolos

ACTIVIDADES

Ejercicio 1

La aplicación WireShark se encuentra ya instalada en los equipos del laboratorio. Por seguridad, los analizadores de protocolos requieren permisos de administrador del sistema para poder realizar capturas del tráfico de la red. Por ello, para que el programa permita la selección de la interfaz sobre la que se van a realizar las capturas a cualquier usuario del sistema, es necesario ejecutar

```
# chmod u+s /usr/bin/dumpcap
```

ya que `dumpcap` es la utilidad de la que depende Wireshark y, de esta forma, hacemos que el proceso `dumpcap` sea propiedad del *root.*

Tras seleccionar la Interfaz sobre la que deberá realizar capturas, arranque una consola y ejecute el comando `ping`. Este comando genera paquetes de tipo ICMP *Request* y ICMP *Reply.* Configure Wireshark para que capture 10 paquetes cualesquiera que circulen por la red, que no capture en modo promiscuo e inicie una captura.

Una vez se han capturado los 10 paquetes, suspenda la ejecución de `ping` (Ctrl-C) y observe los paquetes ICMP recibidos, identificando todos los elementos del protocolo. Responda a las siguientes cuestiones:

- *Datos del paquete*: número, bytes de longitud, protocolos contenidos
- *Trama Ethernet*: Localice únicamente donde se identifica el protocolo de nivel IP que está encapsulado en la zona de datos de la trama Ethernet.
- *Trama IP*: Localice la longitud, origen, destino (¿en qué formato se especifica?) y tipo de protocolo que viaja en la parte de datos: cuál es su código correspondiente?
- *Trama ICMP*: ¿que es ICMP, qué código ("tipo") corresponde a un "Echo Request" y a un "Echo Reply".
- Observe el número de secuencia comparando dos paquetes uno de Request y otro de Reply.

Ejercicio 2

Elimine la limitación de capturas para detener el proceso manualmente y seleccione el modo promiscuo. Arranca una nueva captura. Vuelva a escribir el comando `ping` anterior. Arranque un navegador Internet y realice algunos accesos con él a cualquier dirección web.
Detenga la captura y observe los diferentes tipos de paquetes, contrastando con el ejercicio anterior. En el menú S*tatistics-Protocol Hierarchy* obtendrá una lista porcentual de los diferentes paquetes capturados. Indique cuántos paquetes de cada uno de estos tipos han sido recibidos: ARP_____, IP_____, TCP_____, UDP_____.

Defina un filtro de presentación escribiendo el nombre del protocolo que deseas ver en la barra de filtro. *Filtre* para ICMP y luego para TCP. Es posible ver mensajes dirigidos a otras direcciones de la LAN al captar en modo promiscuo y también mensajes TCP con errores y recuperación de errores (en negro)

Ejercicio 3

El protocolo DNS se utiliza para poder denominar a los computadores mediante nombres simbólicos en lugar de utilizar las direcciones IP más difíciles de recordar. Los servidores DNS se encargan de responder con la dirección IP buscada. El servicio DNS emplea el puerto 53 de UDP.

Realice una captura para estudiar los paquetes generados por la orden

```
$ ping -c 4 www.google.com
```

Examine el contenido del protocolo de la aplicación DNS para la petición de resolución de nombres. Localice el servidor DNS en la primera petición y el contenido de la consulta Query en el cuerpo del mensaje DNS. Eche un vistazo al contenido del datagrama UDP y observe los números de puerto implicados en la transmisión.

Observe que hay dos consultas: la primera obtiene la dirección IP dado un nombre de dominio, la segunda hace lo contrario; localice los parámetros y contraste con el caso anterior. Analice las conversaciones entre diferentes hosts que muestra la ventana *Menú Statistics – Conversations*. Se muestra para protocolos de diferentes niveles Ehernet, IPV (direcciones IP) y UDP. Otras opciones que refieren a conversaciones muestran el mismo tipo de información.

Podemos ver en Wireshark, las secuencia de intercambio de paquetes en un diagrama: *Menú Statistics – Flow Graph*.

Realice de nuevo la misma captura, pero proporcionando la dirección IP de Google al comando `ping` en lugar de su nombre de host y dominio. ¿Será necesario hacer consultas al servicio DNS?

Ejercicio 4

Mediante el comando `telnet`, o `ssh` nos podremos conectar a un usuario de otra máquina de nuestra red. este comando generará tráfico TCP para así poder analizarlo. Será útil designar un filtro como "*host dir_ip and not arp*" siendo *dir_ip* la IP del host en que nos situamos.

Analice las tramas, localice los puertos que conectan en el nivel de transporte y observe el gráfico de flujo como en el caso anterior.

Ejercicio 5

Al igual que con los comandos `telnet, ftp` o `ssh`, el protocolo HTTP también usa TCP en el nivel de transporte. Podemos hacer una petición de una página web desde el navegador a www.google.com o/y www.uah.es y capturar los paquetes generados para identificar algunos parámetros de este protocolo. Será conveniente usar el filtro predefinido "*port 80*" que tiene el nombre "*TCP or UDP port 80 (HTTP)*". Porqué es *port 80*?

En el contenido de los datos http podrá localizar la orden GET / HTTP 1.0 e igualmente podrá localizar el texto de una página html que envía el servidor como respuesta a la petición Get. El seguimiento de las acciones también puede observarse desde *Menu Analyze-Follow TCP stream.*

EVALUACIÓN

Esta práctica no es evaluable.