Universidad Nacional de la Patagonia San Juan Bosco

Facultad de Ingeniería - Departamento de Informática - Sede Trelew.
Cátedra de Redes y Transmisión de Datos.
Curso 2.014.

TP3: Analizador de tráfico.

Introducción

En esta práctica utilizaremos la biblioteca *libpcap* que nos ofrece una interfaz portable para la captura de tramas ethernet . Su capacidad de filtrar el tráfico deseado y trabajar de igual forma con tráfico en tiempo real y tráfico almacenado, lo hace una potente herramienta para la monitorización y el análisis de la dinámica de una red LAN.

Objetivos

- Utilización de la biblioteca *libpcap* para la captura de tramas en una red ethernet y la posterior interpretación de su contenido, estudiando los encapsulados de diferentes protocolos de TCP/IP.
- Adquirir como experiencia el trabajo en grupo.

libpcap

libpcap nos ofrece la posibilidad de capturar los paquetes enviados por la red, con lo que se pueden llevar a cabo procesamiento del tráfico en tiempo real, almacenar lo capturado en un fichero, o procesar lo capturado en un fichero.

Por otro lado, dos modos de escucha son posibles: el modo promíscuo, que permite capturar todos los paquetes que aparezcan en el medio; y no promíscuo donde se capturan sólo los paquetes dirigidos a la interfaz que está utilizando para la captura.

Muchos productos de escucha de paquetes (sniffers) hacen uso de esta biblioteca. Su uso es bastante sencillo, y la estructura general de una aplicación que utilice pcap es como sigue:

– En primer lugar determinamos cuál será la interfaz de la que se capturarán paquetes, bien indicándosela con una cadena de caracteres o bien permitiendo que la seleccione la biblioteca.

– Luego inicializamos libpcap. Se utilizan manejadores para cada interfaz de escucha

– En el caso de que estemos interesados en el filtrado del tráfico se crea un conjunto de reglas de filtrado

– Luego se ejecuta un bucle donde se captura el número de paquetes que se le indique. Cada vez que un paquete cumple la regla que se le ha especificado como filtro, se llama a una función que hemos definido para procesar los paquetes. Existen dos métodos para la obtención de cada trama capturada: mediante la función pcap_next (bloqueante), no devuelve el control al programa hasta que no se captura una trama. Mediante la función pcap_loop (no bloqueante), a la que se le pasa como argumento la referencia a una función que será llamada cada vez que se capture una trama.

– Por último, se cierra la sesión tras haber capturado los paquetes que necesitábamos.

Filtrado.

Antes de comenzar la captura, se puede indicar qué tipo de tráfico se quiere procesar. Para ello, se construye una expresión booleana con reglas que identifiquen los paquetes deseados. Dicha expresión se compone de varias primitivas, que se refieren a un identificador, precedido por uno o varios calificadores.

Los calificadores de tipo (*host*, *net*, *port*) indican la naturaleza del identificador. Con los calificadores de dirección de transferencia se indica si el tráfico que se desea se dirige (*src*) o se origina (*dst*) desde el identificador. Además, los indicadores de protocolo restringen la correspondencia de los paquetes a los que pertenezcan al protocolo indicado. Su valor puede ser uno de los siguientes: *ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp y udp*. Algunos ejemplos: src host redes1, net 172.18/16, tcp port 23.

Existen, por último, algunas primitivas adicionales, como *gateway*, y *broadcast*, y la posibilidad de formar expresiones más complejas mediante *and, or, not*, e incluso expresiones aritméticas .

Práctica a desarrollar. Generalidades.

Haciendo uso de la documentación recomendada, la realización de la práctica consta de la implementación de un analizador en tiempo real que muestre la distribución estadística de los protocolos más destacables del tráfico de la red (Ethernet, ARP, IP, UDP, TCP).

La práctica debe ser desarrollada en el lenguaje de programación C, sobre una plataforma Unix/Linux.

Trabajos a realizar. Obligatorios.

1. Implementación de un analizador de tráfico, del tipo pasivo, que tenga las siguientes opciones:
   i. Muestre la distribución estadística de los protocolos más destacables del tráfico de la red (Ethernet, ARP, RARP, IP, UDP, TCP).
   ii. Que capture y muestre las cabeceras UDP, interpretando los campos.
   iii. Que capture y muestre las cabeceras TCP, interpretando los campos.
   iv. Dado un fichero de trazas, identificar las dos primeras conexiones TCP distintas que encuentre (solo aquellas que comenzaron y finalizaron correctamente).
   v. Que identifique qué aplicaciones, protocolos y máquinas están consumiendo más ancho de banda de red y lo muestre en un listado de mayor a menor (por ejemplo, los 5 primeros).

Las funcionalidades anteriores pueden ser realizadas dandole un fichero de trazas generado por el analizador desarrollado, o por alguna otra aplicación compatible con libpcap como tcpdump o wireshark.

Forma de entrega.

- Archivos fuentes y Makefile. Los fuentes debidamente comentados y respetando las normas del “buen arte”. El entorno de prueba de la cátedra será un Linux en su distribución Ubuntu (versión 12.04).
- Informe con las decisiones de diseño debidamente comentadas (en formato HTML o PDF).
- Informe sobre el protocolo utilizado y bibliografía utilizada (en formato HTML o PDF).

Forma de aprobación

Los items que se tendrán en cuenta para la aprobación del trabajo práctico y los integrantes son los siguientes:

- Funcionamiento de la aplicación desarrollada. La evaluación en este punto se basará en si la funcionalidad cumple con lo solicitado. En caso contrario el trabajo práctico se considerará desaprobado.
- Estructura general de la presentación, su legibilidad y facilidad de lectura y comprensión.
- Contenido del informe y el uso de la información técnica para elaborarlo.
- Evaluación del grupo como un todo y a cada uno de sus integrantes, por lo cual, se exigirá que cada integrante sea capaz de explicar cualquiera de los puntos con los que cuenta el trabajo.

Cuestiones a tener en cuenta

-Los códigos asignados a los protocolos se pueden consultar en /etc/protocols.

-La función getprotobyname puede ser muy útil para identificar qué tipo de protocolo encapsula el datagrama IP capturado.

-En http://www.iana.org/assignments/protocol-numbers se puede encontrar la lista de códigos para protocolos.

-Para comprobar que el analizador funciona correctamente, podemos cotejar los resultados con un analizador ampliamente utilizado, como por ejemplo Wireshark. Primero los capturaremos en un fichero, y después lo pasaremos a cada uno de los programas. De esta forma podremos comprobar que nuestro programa interpreta correctamente la información de las cabeceras.

Acotaciones finales.

- Las mejoras y funcionalidades adicionales que se realicen se tendrán en cuenta en la nota final.
- Para la aprobación del presente trabajo práctico se deben resolver todos los puntos obligatorios (ver **Trabajos a realizar**).
- Los plazos máximo de entrega del presente trabajo práctico son los que figuran en la **Tabla 1**. Luego de dichos plazos el trabajo se considerara desaprobado.

Punto(s)	Fecha
1	1 de julio del 2014

Tabla 1

Bibliografía

-Manual de tcpdump: man tcpdump
-Manual de libpcap: man pcap
-Página de libpcap: http://www.tcpdump.org/pcap.htm
-Páginas de tutoriales para winpcap: http://winpcap.polito.it/docs/man/html/group__wpcap.html
-User Datagram Protocol. RFC 768.
-Transmission Control Protocol, Darpa Internet Program Protocol Specification. RFC 793.
-Internet Protocol, Darpa Internet Program Protocol Specification. RFC 791.