REDES DE COMUNICACIONES I

Práctica 2, sesiones 3/4/5: Disección y filtrado de protocolos

Volver a: Práctica 2, ses...

Objetivo de la práctica

- Entender cómo son las cabeceras de los paquetes que circulan por Internet, y cómo extraer la información que contienen.
- Entender cómo filtrar distintos tipos de tráfico y qué motivación puede haber detrás de esta tarea.

Introducción

Esta segunda práctica consistirá en la realización de un disector de tráfico. Durante la primera y segunda sesión nos centraremos en cómo extraer algunos de los campos de las cabeceras de los paquetes de una traza controlada. En la tercera sesión analizaremos tráfico en vivo y haremos un filtro de captura de paquetes que cumplan con una serie de características, dejando de procesar aquellos paquetes que no las posean.

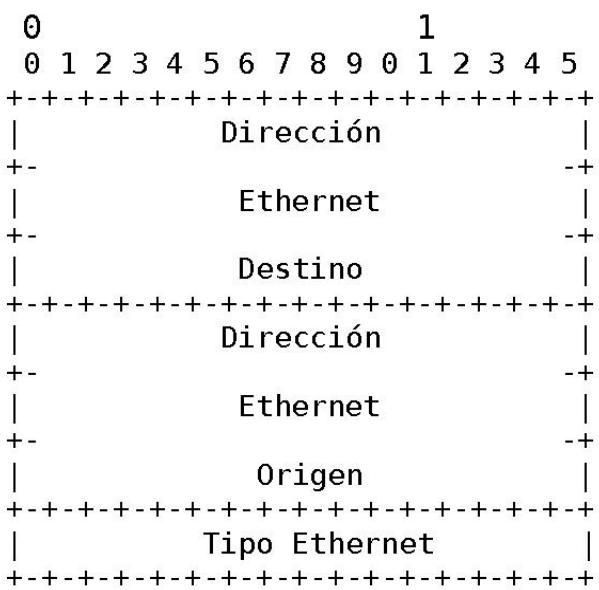
Puede que en este momento no sepa qué significa o cuál es la utilidad de muchos de los campos de las cabeceras que muestra Wireshark. ¡No se inquiete por ello! En esta práctica se pretende aprender a extraer los campos de un protocolo. Cuando vaya avanzando el curso comprenderá el significado concreto y se familiarizará con cada uno de ellos. Note que no estamos ante un escenario extraño, pues a la hora de empezar a inspeccionar tráfico es habitual analizar un protocolo sin conocer en detalle sus campos. Como ayuda, es recomendable recurrir a las definiciones de los distintos protocolos en su documentación, estándar o RFC.

Protocolos a tener en cuenta y sus cabeceras

En la práctica a realizar vamos a suponer que el tráfico de interés es de uno o varios de los siguientes protocolos: Ethernet (nivel 2, de enlace), IP (nivel 3, de red), y TCP o UDP (nivel 4, de transporte). Cualquier otro protocolo no va a ser interpretado. El nivel de aplicación no es relevante.

Podemos obtener la descripción de cabeceras pedidas en sus respectivas RFCs (de aquí en castellano: http://www.rfc-es.org/) o estándares (http://www.ieee802.org/3/):

Ethernet (IEEE 802.3 Ethernet):



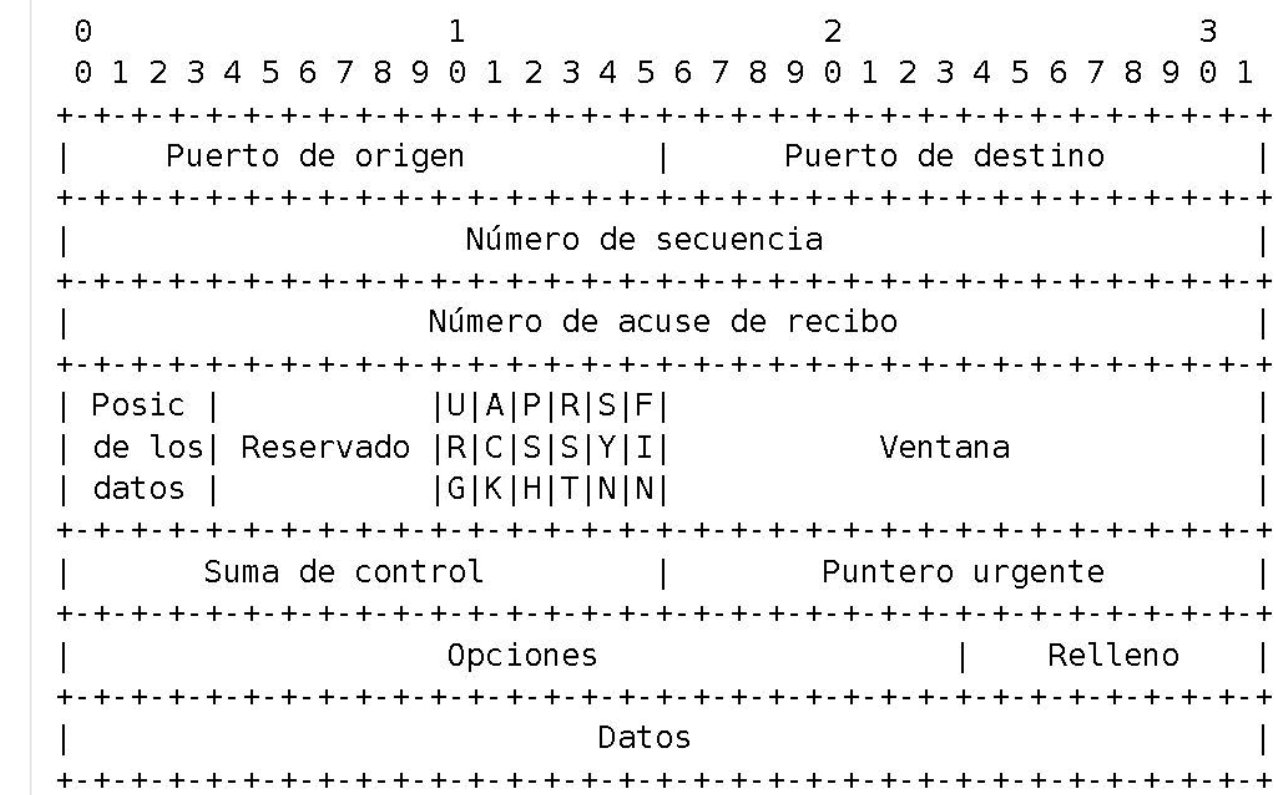
Si el Tipo Ethernet es 0x0800 estamos ante IP. Otros ejemplos son 0x0806 ARP, 0x0842 Wake-on-Lan, etc. (http://standards.ieee.org/develop/regauth/ethertype/eth.txt)

RFC 791, IP:

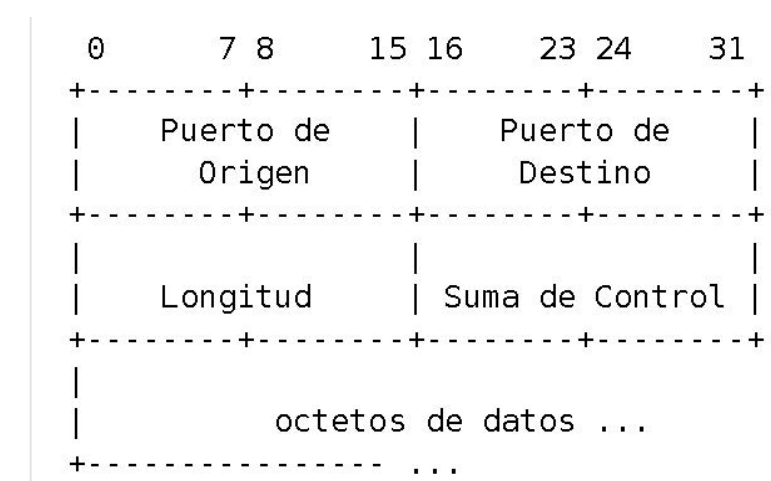
Donde Protocolo vale 6 para TCP, y 17 para UDP. Otros ejemplos son 1 para ICMP, o el 0 que está reservado (RFC 791).

Los campos Opciones y Relleno son optativos, típicamente no suelen aparecer. El campo IHL nos indicará el tamaño de la cabecera que permite indirectamente saber el tamaño del campo opciones.

RFC 793, **TCP:**



RFC 768, **UDP:**



Si desea profundizar en los protocolos, deberá consultar los RFCs, standards, el libro de teoría o preguntar al profesor.

Detalles de implementación

Repase la lección "Comandos útiles para programación" (sobre todo, las secciones "Operaciones con arrays binarios", "Operaciones a nivel de bit" y "Orden de red y de host"). Esta última sección explica las funciones htons(·)/htonl(·),ntohs(·)/ntohl(·) que son de uso muy aconsejable para lidiar con el problema del "endianness".

Aunque existen definiciones de tipos de estructuras que por defecto "parsean" las cabeceras (por ejemplo struct ether_header), no las use, pues aunque útiles, vamos a realizar esta trabajo "a mano" con objetivos didácticos. **Una práctica que los use se valorará como cero.**

Por el mismo motivo **NO** use filtros BPF (por ejemplo funciones pcap_compile, pcap_filter). **Una práctica que los use se valorará como cero.**

Se facilita un ejemplo descargable en Moodle: practica2.c. En este ejemplo se muestra como empezar esta práctica pues muestra una serie de campos ya *diseccionados*. Adicionalmente se muestra cómo "*parsear*" argumentos de entrada en C aunque esta parte es sólo una propuesta de implementación.

En el ejemplo proporcionado se utiliza la función *pcap_next_ex*. **Una práctica que utilice una función diferente (pcap_loop/pcap_next) se valorará como cero.**

Finalmente, para generar trazas con paquetes variados respecto a los protocolos de nivel 2 y 3 con objeto de realizar pruebas use comandos como arping, ping (cambiando el tamaño), hping3 o traceroute.

Ejercicio

Entregue los fuentes (.c y .h) y **makefile** usados para implementar un programa C basado en libpcap que:

1. Por cada paquete muestre por pantalla:
   - Los campos de nivel 2 (con formato hexadecimal y 2 dígitos por byte): ambas direcciones Ethernet (separando cada byte por ":") y el tipo de protocolo que encapsula (puede ponerlo en hexadecimal (sin :) o lo puede indicar con palabras). En caso de que el siguiente protocolo no sea IPv4, se deberá indicar por pantalla que no es el protocolo esperado y no se imprimirá la información correspondiente a los siguientes niveles.
   - Los campos de nivel 3 en decimal: versión IP, longitud de cabecera, longitud total, posición/desplazamiento (**!!**), tiempo de vida, protocolo, y ambas direcciones IP (estas en el formato "decimal con punto" x.x.x.x). **(NO imprima ni extraiga el campo Identificación pues no se pide).**
     - Hay un ejemplo en Moodle de una traza con paquetes con fragmentación pues el campo posición/desplazamiento es levemente menos simple.
     - Si el campo posición/desplazamiento es distinto de 0, NO se deben analizar los campos del siguiente nivel. En este caso se nos indica que el paquete IP leído no es el primer fragmento, de forma que no contendrá la cabecera de nivel 4. En sentido estricto, sería necesario proceder a la unión de todos los fragmentos IP para proceder al análisis del segmento de nivel 4, pero por simplicidad solamente atenderemos al primero que es el que contiene los campos que nos interesan.
     - En caso de que el siguiente protocolo no sea ni TCP ni UDP, se deberá indicar por pantalla que no es el protocolo esperado y no se imprimirá la información correspondiente a los siguientes niveles.
   - Los campos de nivel 4 en decimal: número de puerto de origen y puerto de destino tanto UDP como TCP. En el caso UDP también debe mostrarse el campo longitud en decimal. En el caso de TCP también deben

mostrarse los valores de las banderas SYN y ACK.

2. Finalmente queremos implementar una funcionalidad de filtrado:
    - Debe seguir el siguiente formato de ejecución: practica2 <**-f ficheroPcap / -i interfaz>** [-ipo IP_Origen] [-ipd IP_Destino] [-po Puerto_Origen] [-pd Puerto_Destino].
        - Importante, debido a los corchetes [⋯], entenderemos que los parámetros son opcionales. Por lo tanto, los parámetros pueden no estar presentes y encontrarse en cualquier orden.
        - Los parámetros -f y -i son excluyentes, esto es, no pueden encontrarse simultáneamente. No obstante, al menos uno de los dos es obligatorio.
    - Buscamos respetar el análisis capa-a-capa de la pila de protocolos de red. Por lo tanto, **la detección de filtros se producirá en el momento que se imprima por pantalla el campo sobre el que se ejecuta el filtro**. Esto implica, que los bytes del paquete solo serán recorridos una vez. Se penalizará aquellas prácticas que comprueben los filtros y descarten los paquetes antes de imprimir el valor del campo en cuestión.

Entrega

Denomine a los ficheros de entrega practica2.c/practica2.h. Añada un fichero **leeme.txt** que incluya los nombres de los autores, comentarios que se quieran transmitir al profesor y, en caso de entregar algún fichero más la explicación del mismo. **No olvide el makefile**.

Comprima en un zip todo lo que vaya a entregar y llámelo practica2_YYYY_PXX.zip, donde YYYY es el grupo al que pertenece (1301,1302,etc), y XX (y solo XX) es el número de pareja (con dos dígitos). Si no recuerda su número de pareja, mírelo en el "Listado grupos de trabajo (parejas)".

Solo es necesario que suba la entrega un miembro de la pareja. En caso de dudas pueden subir ambos miembros la práctica.

Criterios de evaluación

Ejercicio: Entrega el día 19 de octubre hasta las 23:55.

- Información nivel 2: 10%
- Información nivel 3: 20%
- Campo desplazamiento/offset: 20%
- Información nivel 4: 20%
- Captura desde interfaz de red: 5%
- No analizar tráfico cruzado (que no sea IP|TCP o IP|UDP): 10%
- Filtrado de paquetes: 15%
- **Mostrar campos no pedidos: -5% por cada nivel en el que se muestren.**

Cuestionario: El día 6 de Octubre se realizará un breve cuestionario individual en Moodle. El objetivo del cuestionario será afianzar los objetivos de la práctica por parte de los estudiantes. Se considera que para entonces debe haberse realizado al menos la impresión de campos de nivel 2 y 3.

Control individual: Control sobre la práctica el día 20 de Octubre.

Importante, el control planteará ejercicios de modificación del código de la práctica, esto implica, primero, que los alumnos deben traer consigo sus prácticas hechas el día del control, y segundo, conocerlas bien como para ser capaz de modificar aspectos en un tiempo limitado. Las cabeceras en cualquier caso estarán accesibles.
Las modificaciones pueden consistir en extraer nuevos campos, o simples cambios/añadidos en la pila de protocolos (esto es, en las cabeceras). Puede ser útil que su programa imprima el número de paquete que está analizando (respecto al total de la traza).
No se permitirá el uso de Wireshark. No olvide ser puntual al control pues empezará a "y 5".
Última modificación: viernes, 29 de septiembre de 2017, 12:03

Volver a: Práctica 2, ses...