SNIFFER

Objetivo: Identificar los campos que conforman un datagrama IP y analizar su comportamiento.

Desarrollo:

Utilizando lenguaje C para Linux, implemente un sniffer que capture los paquetes que están viajando en la red, específicamente se deben capturar tramas Ethernet. Se debe verificar que la carga útil de la trama capturada es un datagrama IPv4 (0x800), en caso de que no se cumpla esta condición, la trama es ignorada y se procesa la siguiente. Si se cumple con la condición anterior, se debe analizar la trama capturada y extraer la siguiente información:

- *Dirección IP fuente*
- *Dirección IP destino*
- *Longitud de cabecera en bytes*
- *Longitud total del datagrama IP en bytes*
- *Identificador del datagrama*
- *Tiempo de vida*
- *Protocolo de capa superior*

Además, se debe determinar:

- *Longitud de carga útil*
- *Tipo de servicio utilizado*
- *Si el datagrama está fragmentado o no*
- *Número de fragmento (único, primero, intermedio o último)*
- *Primer y último byte que contiene el datagrama*

Para identificar el protocolo de capa superior se debe considerar que:

- ICMPv4 tiene un valor en el campo protocolo 0x01
- IGMP tiene un valor en el campo protocolo 0x02
- IP tiene un valor en el campo protocolo 0x04
- TCP tiene un valor en el campo protocolo 0x06
- UDP tiene un valor en el campo protocolo 0x11
- IPv6 tiene un valor en el campo protocolo 0x29
- OSPF tiene un valor en el campo protocolo 0x59

Finalmente, se deben mostrar las siguientes estadísticas:

- Número de paquetes capturados de cada uno de los protocolos de capa superior.
- Número de paquetes por cada dirección IP diferente, especificando claramente cuántos de esos paquetes fueron transmitidos y cuántos fueron recibidos.
- Número de paquetes según su tamaño:
  - ✓ 0-159
  - ✓ 160-639
  - ✓ 640-1279

- ✓ 1280-5119
- ✓ 5120 o mayor

Toda esta información debe ser almacenada en un archivo de texto, el sniffer debe ser programado para poder dar como dato de entrada el número de paquetes que deben ser capturados mediante una cadena de entrada.

Recuerden que para poder implementar el sniffer es necesario utilizar un socket de capa de enlace de datos (*SOCK_RAW*), la familia de protocolos a utilizar debe ser PF_PACKET y se debe configurar la tarjeta de red en *"modo promiscuo"* para que reciba todos los paquetes que están viajando en la red y no solamente los que van dirigidos a ella; para esto se debe utilizar la función *ioctl()* y la estructura *struct ifreq,* dicha estructura contiene la información de los adaptadores de red, y la función *ioctl()* permite modificar algunos parámetros de estos adaptadores, para poder utilizarlas es necesario incluir las librerías <*sys/ioctl.h*> y <*net/if.h*>. Una vez que se cierra el sniffer es necesario restablecer los valores originales de la tarjeta de red, esto se puede logar con la siguiente instrucción:

system("/sbin/ifconfig nombre_adaptador -promisc")