TP 1-B

Integrantes:

- Cardinale Ignacio
- delValle Facundo
- Dimperio Bautista
- Ratcliffe Patricio.

1- Utilizando el código raw.c como base escribir un "sniffer" que es un programa que muestra el contenido del tráfico que llega.

Primero creamos el raw socket con el protocolo TCP:

```
// Crear socket raw
if ((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_TCP)) < 0) {
    perror("socket");
    exit(EXIT_FAILURE);
}
```

Lo configuramos para que reciba todos los paquetes que le llegan:

```
50
51      // Recibir paquetes
52      while (1) {
53          int bytes_recibidos = recvfrom(sockfd, buffer, sizeof(buffer), 0, NULL, NULL);
54          if (bytes_recibidos < 0) {
55              perror("recvfrom");
56              exit(EXIT_FAILURE);
57          }
58
59          procesar_paquete(buffer, bytes_recibidos);
60      }
```

Después a cada paquete que llegaba lo llevamos a un función el cual captura el encabezado (las primeras dos líneas de código) y después compara este encabezado con el protocolo correspondiente, IPPROTO_TCP para TCP, IPPROTO_UDP para UDP, IPPROTO_ICMP para ICMP y dependiendo que tipo de paquete era que tenga una determinada salida:

```
void procesar_paquete(unsigned char *buffer, int size) {
    struct iphdr *encabezado_ip = (struct iphdr *)buffer;
    unsigned short longitud_encabezado_ip = encabezado_ip->ihl * 4;

    if (encabezado_ip->protocol == IPPROTO_TCP) {
        struct tcphdr *encabezado_tcp = (struct tcphdr *)(buffer + longitud_encabezado_ip);
        unsigned int puerto_origen = ntohs(encabezado_tcp->source);
        unsigned int puerto_destino = ntohs(encabezado_tcp->dest);

        printf("Paquete TCP - Puerto de origen: %u, Puerto de destino: %u\n", puerto_origen, puerto_destino);
        // Imprimir los datos del paquete TCP
        printf("\n");
    } else if (encabezado_ip->protocol == IPPROTO_UDP) {
        struct udphdr *encabezado_udp = (struct udphdr *)(buffer + longitud_encabezado_ip);
        unsigned int puerto_origen = ntohs(encabezado_udp->source);
        unsigned int puerto_destino = ntohs(encabezado_udp->dest);

        printf("Paquete UDP - Puerto de origen: %u, Puerto de destino: %u\n", puerto_origen, puerto_destino);
        // Imprimir los datos del paquete UDP
        printf("\n");
    } else if (encabezado_ip->protocol == IPPROTO_ICMP) {
        printf("Paquete ICMP\n");
    } else {
        printf("Paquete de protocolo desconocido\n");
    }
}
```

La salida al ejecutar el código es:

```
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58514
Paquete TCP - Puerto de origen: 58514, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 58514, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58514
Paquete TCP - Puerto de origen: 58516, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58516
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58514
Paquete TCP - Puerto de origen: 58514, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 58516, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58516
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58516
Paquete TCP - Puerto de origen: 58516, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58516
Paquete TCP - Puerto de origen: 58516, Puerto de destino: 40553
```

2- Enviar tráfico al "sniffer" desde el cliente escrito en la parte A del TP1

Para esta parte usando el código del cliente de la parte A le mando por el puerto 7801 paquetes TCP y aparece de la forma:

```
Paquete TCP - Puerto de origen: 42260, Puerto de destino: 7801
Paquete TCP - Puerto de origen: 7801, Puerto de destino: 42260
Paquete TCP - Puerto de origen: 42260, Puerto de destino: 7801
Paquete TCP - Puerto de origen: 42260, Puerto de destino: 7801
Paquete TCP - Puerto de origen: 7801, Puerto de destino: 42260
Paquete TCP - Puerto de origen: 7801, Puerto de destino: 42260
Paquete TCP - Puerto de origen: 42260, Puerto de destino: 7801
Paquete TCP - Puerto de origen: 42260, Puerto de destino: 7801
Paquete TCP - Puerto de origen: 7801, Puerto de destino: 42260
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58514
Paquete TCP - Puerto de origen: 58514, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 58516, Puerto de destino: 40553
Paquete TCP - Puerto de origen: 40553, Puerto de destino: 58516
```

3- Enviar tráfico ICMP al "sniffer" y mostrar los resultados del LOG con comentarios.

Para esta parte es igual a lo anterior lo único que cambia es el protocolo que vamos a usar para crear el socket, donde antes era:

```
// Crear socket raw
if ((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_TCP)) < 0) {
    perror("socket");
    exit(EXIT_FAILURE);
}
```

Ahora es:

```
45      // Crear socket raw
46      if ((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) < 0) {
47          perror("socket");
48          exit(EXIT_FAILURE);
49      }
```

Y en el cliente al crear el socket se crea también con el protocolo ICMP:

```
    // Crear socket raw ICMP

    if ((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) < 0) {
        perror("socket");
        exit(EXIT_FAILURE);
    }
```

Ahora en la consola cuando yo mando un paquete ICMP desde el cliente en la terminal de clienta aparece:

```
nacho@DESKTOP-NP5GCRK:~$ sudo ./cliente 127.0.0.1
Paquete ICMP enviado al servidor 127.0.0.1
nacho@DESKTOP-NP5GCRK:~$
```

Y en la del servidor al llegar un paquete ICMP aparece como:

```
nacho@DESKTOP-NP5GCRK:~$ sudo ./sniffer
Paquete ICMP
```