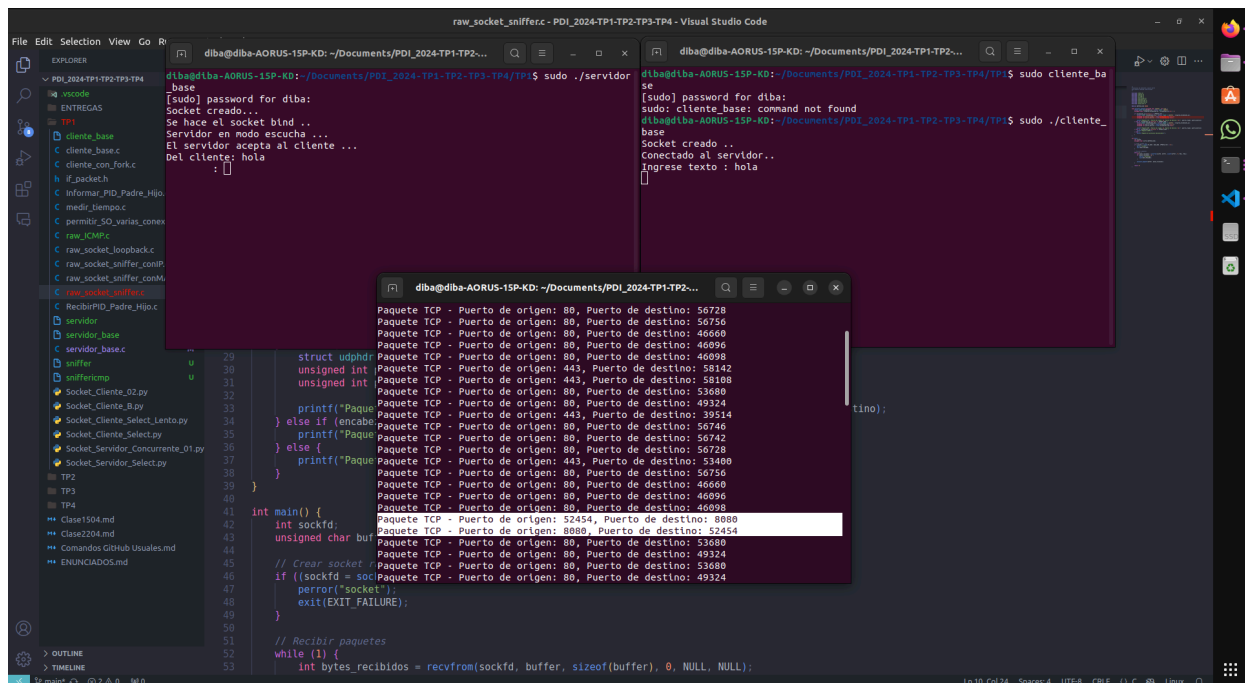


Capon, Ian; Reno, Franco; Di Bartolo, Ignacio; Bachman, Ivan.

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

Utilizamos el código de `raw_socket_sniffer.c` para el inciso 2, modificado para escanear paquetes ICMP en el inciso 3.

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



```
diba@diba-AORUS-15P-KD: ~/Documents/PDI_2024-TP1-TP2-...
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56728
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56756
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46660
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46096
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46098
Paquete TCP - Puerto de origen: 443, Puerto de destino: 58142
Paquete TCP - Puerto de origen: 443, Puerto de destino: 58108
Paquete TCP - Puerto de origen: 80, Puerto de destino: 53680
Paquete TCP - Puerto de origen: 80, Puerto de destino: 49324
Paquete TCP - Puerto de origen: 443, Puerto de destino: 39514
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56746
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56742
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56728
Paquete TCP - Puerto de origen: 443, Puerto de destino: 53400
Paquete TCP - Puerto de origen: 80, Puerto de destino: 56756
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46660
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46096
Paquete TCP - Puerto de origen: 80, Puerto de destino: 46098
Paquete TCP - Puerto de origen: 52454, Puerto de destino: 8080
Paquete TCP - Puerto de origen: 8080, Puerto de destino: 52454
Paquete TCP - Puerto de origen: 80, Puerto de destino: 53680
Paquete TCP - Puerto de origen: 80, Puerto de destino: 49324
Paquete TCP - Puerto de origen: 80, Puerto de destino: 53680
Paquete TCP - Puerto de origen: 80, Puerto de destino: 49324
```

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

Modificamos levemente el archivo `raw_socket_sniffer.c` para capturar tráfico ICMP. Esto lo logramos reemplazando `IPPROTO_TCP` por `IPPROTO_ICMP`. Luego le hicimos ping al localhost para poder ver al sniffer en acción.

