Tarea 3: Inyectando paquetes UDP en otro socket
Redes
Plazo de entrega: 12 de julio 2023

José M. Piquer

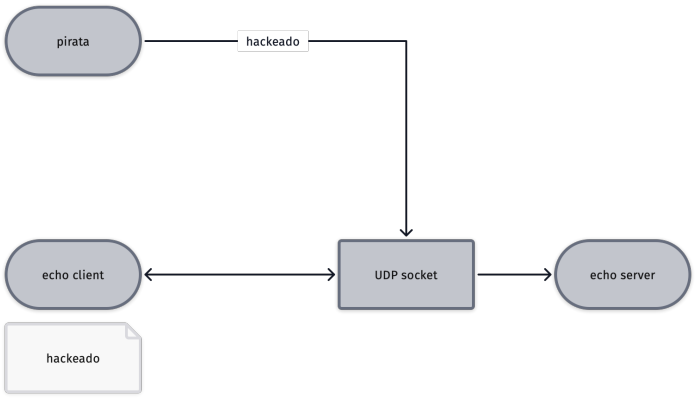

Fig. 1: Esquema general

1. Descripción

Su misión, en esta tarea, es inyectar un mensaje "pirata" en un socket UDP. Usaremos de ejemplo el cliente y servidor de eco visto en clases. Con un cliente enviando paquetes para tener eco desde el servidor, Uds deben generar desde otro proceso, en el mismo computador del cliente, un paquete UDP que le llegue al servidor y este conteste con su eco habitual. Pero el paquete que Uds envían debe estar construido de tal forma que venga con los headers modificados para que ese eco en realidad le llegue al cliente "normal"

que estaba enviando paquetes de antes.

Para esto se les pide usar *scapy*, un sistema para manipular a mano paquetes de red. Uds deben "falsificar" un paquete UDP para que el servidor crea que viene del cliente y así lo responda. Scapy se puede usar directamente con un script o pueden usar el módulo para Python que permite utilizarlo desde dentro de un programa.

Para lograr que el paquete le llegue al cliente existente, necesitamos una información extra: el número de puerto UDP que el cliente está usando en su extremo. El del servidor lo conocemos, es el 1818 de siempre. Pero el del cliente es aleatorio y cambia para cada cliente. Para esto, les entregamos un cliente y un servidor modificados. El cliente escribe en su salida el número de puerto asignado. El servidor muestra en pantalla todo lo que recibe. Una versión de ese servidor está corriendo en anakena en el puerto 1818 para que lo usen en sus pruebas.

HINTS:

- Windows: Además de instalar scapy, deben instalar npcap
- Linux: para poder probar en 127.0.0.1, deben agregar la línea:

```
conf.L3socket=L3RawSocket
```

- MacOSX: Todo funciona, pero no agreguen la línea anterior, que hace que nada más funcione!

2. Entregables

Básicamente entregar el archivo con el pirata que logra el objetivo y un archivo que explique cómo usarlo para inyectar un paquete al servidor en anakena. Deben lograr que el cliente normal de eco reciba de vuelta del server un paquete que diga "hackeado". Explique en detalle el sistema operativo y el ambiente en que probó la tarea para poder replicarlo.

En un archivo aparte responder las preguntas siguientes (digamos, unos 5.000 caracteres máximo por pregunta):

1. Intente usar el script que inventó ahora con IPv6: ¿Cuánto se debe modificar lo hecho? ¿Por qué? (si logra que funcione en `::1` con IPv6, le daremos un punto más en esta tarea).

2. Este ejercicio muestra lo trivial que es la seguridad en UDP. Si uno quisiera tener un sistema seguro en UDP, ¿cómo podríamos protegernos de este tipo de falsificaciones?

3. En TCP, ¿sería igual de trivial inyectar un paquete de datos?

4. Si ahora queremos que el pirata esté en otro computador que el cliente, ¿se podría hacer lo mismo con scapy? ¿Cómo?