

Tarea 3 Redes

1. Para que funcione con IPv6 en el código habría que cambiar la forma en la que se escribe el paquete, cambiando:

```
packet = IP(src=client_ip, dst=server_ip) / UDP(sport=client_port, dport=server_port) / Raw(load=message.encode())
#se cambia a lo siguiente
packet = IPv6(src=client_ip, dst=server_ip) / UDP(sport=client_port, dport=server_port) / Raw(load=message.encode())
```

Esto se hace porque se necesita un paquete del tipo IPv6. El resto del código se puede mantener, ya que UDP no se ve afectado por el cambio a IPv6. Eso sí, es posible que no funcione ya que IPv6 es en teoría más seguro que IPv4.

2. Para solucionar la vulnerabilidad que se nos presenta, una buena opción sería aplicar un método de encriptación, esto se podría lograr de varias maneras, unas de esas sería haciendo una autenticación de mensajes (MAC) o añadiendo una firma digital, esta última es más poderosa ya que además de autenticar el mensaje también garantiza la identidad del emisor del mensaje.
3. No, inyectar un paquete de datos en una conexión TCP no es trivial. TCP como es un protocolo cuya finalidad es la conexión y confiable, garantiza que los datos se entreguen de manera ordenada y sin errores.

En una situación normal los paquetes TCP están secuenciados y verificados mediante el mecanismo de números de secuencia y sumas de verificación. Así que cualquier intento de inyectar un paquete de datos no válido en una conexión TCP se detectaría como algo extraño, lo cual provocaría que se termine por descartar.
4. Como pudimos darnos cuenta durante la tarea, Scapy como herramienta es super poderosa. Además, este nos permitiría lograrlo, pero para esto tendríamos que lograr modificar la dirección IP de origen y así parezca que viene del cliente. Ahora, el problema sería lograr modificar esta dirección IP, pero en principio sí se podría.

Tomas Alvarado Pizarro