**Informe del Trabajo Práctico N.º 4**

**Parte A:**

**Capa de Transporte - Sockets TCP y UDP - Puertos**

**Introducción**

Este trabajo práctico tuvo como objetivo principal aplicar los conocimientos de la **capa de transporte** del modelo OSI, trabajando tanto con sockets **UDP como TCP**, a través del desarrollo de aplicaciones de chat entre computadoras. También se incorporó el uso de herramientas de análisis de tráfico como **Wireshark**, y escáneres de seguridad como **Nmap** y **Nikto**, para analizar el comportamiento de las conexiones, la exposición de puertos y detectar vulnerabilidades en la red. El trabajo se organizó en cuatro actividades, combinando programación, captura de tráfico real y análisis de seguridad.

**Actividad 1: Sockets UDP (Chat sin servidor)**

Se desarrolló una aplicación de chat utilizando **sockets UDP** y **broadcast**, en la que cada instancia del programa puede enviar y recibir mensajes desde cualquier otra computadora de la LAN.

**Características principales:**

* Cada usuario ingresa su nombre al iniciar.
* El mensaje se envía a la IP de broadcast (255.255.255.255) en el puerto 60000.
* Los mensajes recibidos se muestran en el formato:  
  usuario (IP) dice: mensaje
* Al ingresar exit, se notifica a todos los usuarios que el participante abandonó la conversación.
* Al iniciar un nuevo proceso, se notifica su ingreso con usuario:nuevo.

**Ejecución del código:**

**python3 act.py**

**Actividad 2: Sockets TCP (Chat cliente-servidor)**

Se implementó una aplicación de chat mediante sockets TCP con comunicación **cliente-servidor**. El servidor permanece activo y acepta múltiples conexiones de clientes.

**Funcionalidad del servidor:**

* Escucha en el puerto 60000.
* Acepta conexiones simultáneas mediante hilos.
* Reenvía los mensajes recibidos a todos los demás clientes conectados.
* Muestra mensajes de conexión y desconexión.

**Funcionalidad del cliente:**

* Se conecta al servidor mediante su IP.
* Envía el nombre de usuario al conectarse.
* Envía mensajes al servidor y recibe los mensajes del resto de los usuarios.
* Finaliza la conexión al escribir exit.

**Ejecución del código:**

Servidor:

**python3 act2Server.py**

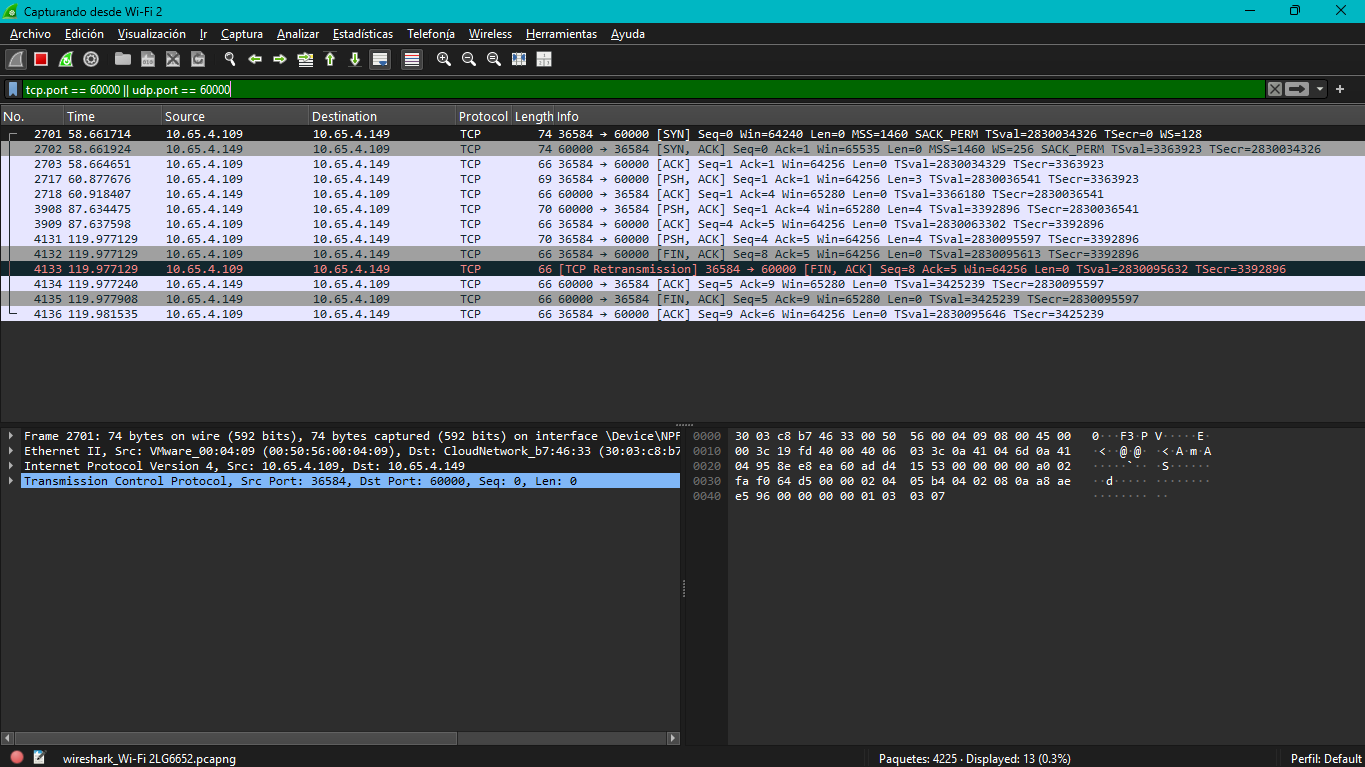
Cliente:

**python3 act2Cliente.py**

**Actividad 3: Análisis de tráfico (Wireshark)**

Se utilizó **Wireshark** para analizar el tráfico generado por las aplicaciones de las Actividades 1 y 2.

* Se aplicaron los filtros udp.port == 60000 y tcp.port == 60000 para observar el tráfico.
* Se verificó que los mensajes eran visibles en los paquetes transmitidos.
* Se confirmó que **los datos transmitidos no están cifrados**, por lo tanto, **un intruso podría ver el contenido de los mensajes** si captura los paquetes.



**Actividad 4: Escaneo de puertos y vulnerabilidades**

**4.2: Análisis de puertos peligrosos en un equipo compañero**

Se escaneó la IP **10.65.4.110** en busca de puertos comúnmente explotables.

**Comando usado:**

sudo nmap -p 21,22,23,25,135-139,443,445,3389 10.65.4.110

**Resultado:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Puerto** | **Estado** | **Servicio** | **Riesgo** |
| 22 | Abierto | SSH | Si no está bien configurado, puede ser vulnerable a ataques por fuerza bruta o accesos remotos indebidos. Se recomienda usar autenticación por clave pública, desactivar root y aplicar control de acceso por IP. |

**4.3 – Escaneo con Nikto**

Se realizó un escaneo de 100 hosts aleatorios buscando servidores con puerto 80 abierto. Uno de los hosts detectados fue:

* **IP:** 23.106.202.158
* **Ubicación geográfica:** Seattle, EE. UU.
* **Servidor Web:** Microsoft-IIS/7.5

**Vulnerabilidades detectadas:**

1. **ETags filtrados:** Riesgo bajo – permite fingerprinting del servidor.
2. **Falta de X-Frame-Options:** Riesgo moderado – susceptible a clickjacking.

**4.4 – Escaneo de IPs de Corea del Norte**

Se escaneó el rango 175.45.176.0/22 correspondiente a Corea del Norte.

**Comando usado:**

sudo nmap -p 80 --open 175.45.176.0/22

**Resultado:**

* **Cantidad de IPs con puerto 80 abierto:** 11
* **IPs detectadas:**
  + 175.45.176.69
  + 175.45.176.71
  + 175.45.176.75
  + 175.45.176.76
  + 175.45.176.80
  + 175.45.176.81
  + 175.45.176.85
  + 175.45.176.91
  + 175.45.177.1
  + 175.45.177.10
  + 175.45.177.11

**Conclusión**

Este trabajo permitió comprender de forma práctica cómo funcionan las comunicaciones de red en la **capa de transporte**, tanto usando **UDP sin conexión** como **TCP orientado a conexión**. Además, se exploraron aspectos clave de la **seguridad de red**, como el análisis de tráfico no cifrado, escaneo de puertos potencialmente inseguros, y detección de vulnerabilidades en servidores web. Herramientas como **Wireshark, Nmap y Nikto** fueron fundamentales para estas tareas.