

TP 4 REDES

Integrantes : Lautaro Larosa

Ejercicio 1

Descripción:

Este código implementa un **chat en tiempo real usando UDP con transmisión por broadcast** (difusión a todos los dispositivos en la red local). Las características principales son:

- Permite a múltiples usuarios unirse al chat usando un nombre.
- Detecta cuando un usuario se conecta (**nuevo**) o abandona (**exit**).
- Usa **hilos separados** para enviar y recibir mensajes simultáneamente.
- Funciona en una red local (LAN) gracias al broadcast (**255.255.255.255**).

Cómo Ejecutarlo:

1. Requisitos:

- Python 3 instalado.
- Varias computadoras en la **misma red local** (o varias terminales en la misma máquina con direcciones IP distintas para pruebas).

2. Pasos:

- Copia el código en un archivo (ej. **chat_udp.py**).
- Ejecuta el script en cada computadora que participará en el chat:

```
python chat_udp.py
```

Cuando se solicite, ingresa un **nombre de usuario**.
Escribe mensajes (para salir, escribe `exit`).

Ejercicio 2

Código 1: Servidor TCP Multihilo

Descripción:

Este script implementa un **servidor TCP** que maneja múltiples clientes simultáneamente usando hilos. Las características clave son:

- **Gestión de conexiones:** Acepta clientes en el puerto `5000` y crea un hilo por cada uno.
- **Broadcast manual:** El servidor puede enviar mensajes a todos los clientes conectados.
- **Sincronización segura:** Usa un **Lock** (`threading.Lock`) para evitar condiciones de carrera al modificar la lista de clientes.
- **Control de salida:** El servidor no se cierra si hay clientes activos (a menos que se fuerce con `Ctrl+C`).

Cómo Ejecutarlo:

1. Requisitos:

- Python 3 en la máquina que actuará como servidor.
- Firewall configurado para permitir conexiones en el puerto `5000`.

2. Pasos: `python servidor_tcp.py`

1-El servidor iniciará en `0.0.0.0:5000` y esperará conexiones.

2-Para enviar mensajes a todos los clientes, escribe texto en la terminal del servidor.

3-Para cerrar el servidor, escribe `exit` (solo si no hay clientes conectados).

Código 2: Cliente TCP

Descripción:

Este script es el **cliente** que se conecta al servidor TCP. Sus funcionalidades incluyen:

- **Conexión a un servidor:** Solicita la IP del servidor para conectarse al puerto `5000`.
- **Recepción en segundo plano:** Un hilo separado recibe mensajes del servidor sin bloquear la entrada del usuario.
- **Envío de mensajes:** El usuario puede escribir mensajes o `exit` para desconectarse.

Cómo Ejecutarlo:

1. Requisitos:

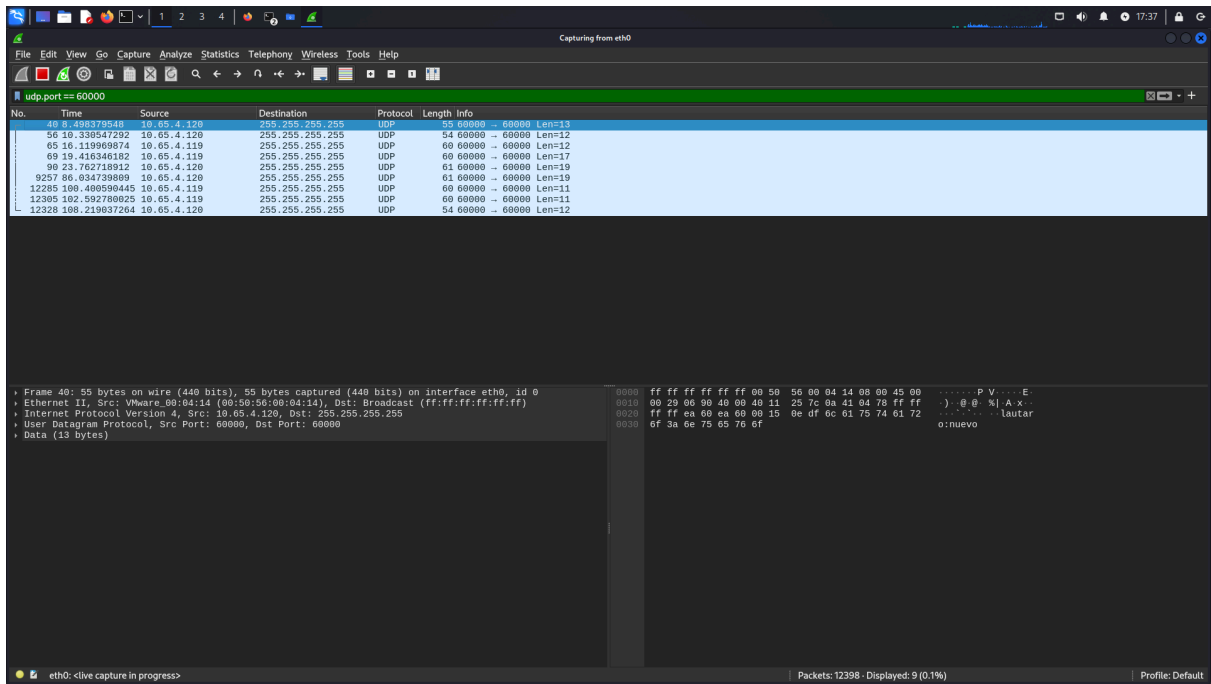
- Conocer la IP del servidor (ejemplo: `192.168.1.1`).

2. Pasos:

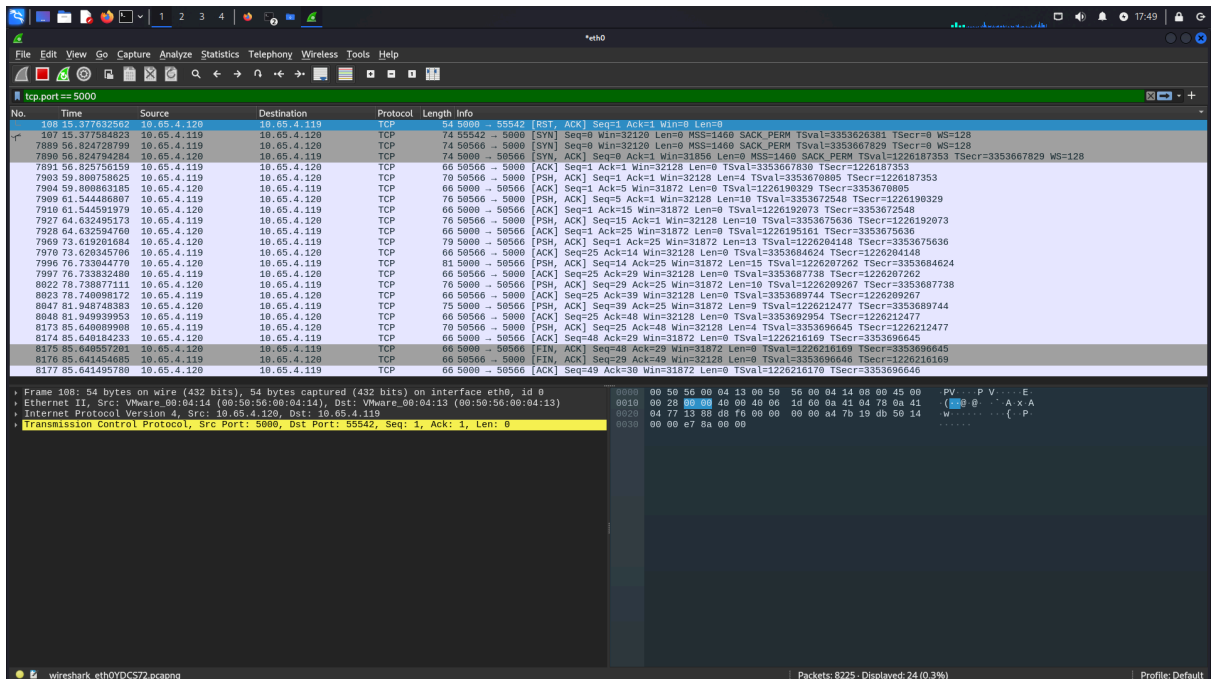
```
python cliente_tcp.py
```

- Ingresa la IP del servidor cuando se solicite.
- Escribe mensajes (o `exit` para salir).

Ejercicio 3



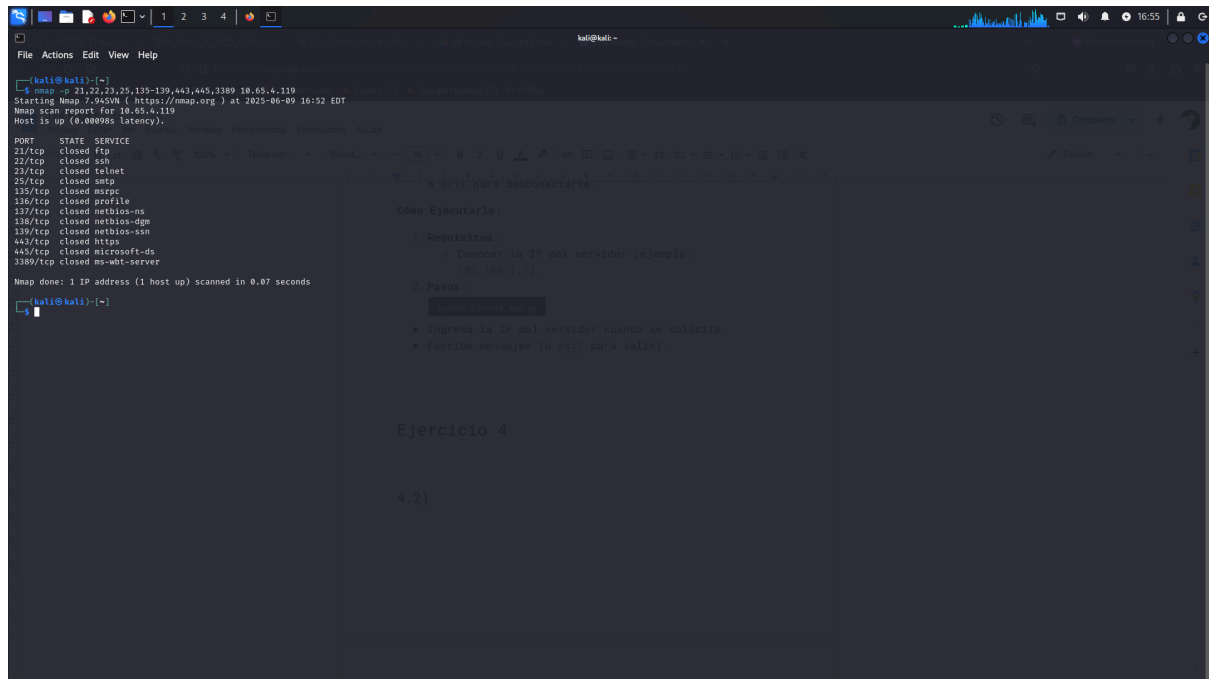
Se capturaron paquetes UDP enviados desde la IP 10.65.4.120 y desde la 10.65.4.119 usando el puerto 60000



Se observan paquetes TCP con secuencias (Seq), confirmaciones (Ack), tamaños (Len) y timestamps (TSval, TSecr).

Ejercicio 4

4.2)



```
kali@kali:~$ nmap -sS 21,22,23,25,135-139,443,445,3389 10.4.65.119
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-09 16:52 EDT
Nmap scan report for 10.4.65.119
Host is up (0.0000ms latency).
Not shown: 65535 closed ports
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    closed ssh
23/tcp    closed telnet
25/tcp    closed smtp
135/tcp   closed msrpc
136/tcp   closed profile
137/tcp   closed netbios-ns
138/tcp   closed netbios-dgm
139/tcp   closed netbios-ssn
443/tcp   closed https
445/tcp   closed microsoft-ds
3389/tcp  closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds

kali@kali:~$
```

Como Ejecutarlo:

1. Requisitos:
 - Conocer la IP del servidor (ejemplo: 192.168.1.1).
2. Pasos:
 - Ejecutar el comando: `nmap -sS 21,22,23,25,135-139,443,445,3389 10.4.65.119`
 - Ingresar la IP del servidor cuando se solicite.
 - Escribir mensajes [o exit para salir].

Se utilizo la computadora de la facultad con ip: 10.4.65.119. Podemos ver que ninguno de los puertos solicitados está abierto por lo tanto la computadora del compañero no presenta vulnerabilidades.

4.3)

Ejecutamos: `nmap -iR 100 -p 80 --open`

```

Starting Nmap 7.92 ( https://nmap.org ) at 2025-05-17 22:46 -03
Nmap scan report for ec2-52-202-2-227.compute-1.amazonaws.com (52.202.2.227)
Host is up (0.22s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for tk2-238-28520.vs.sakura.ne.jp (160.16.124.24)
Host is up (0.56s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 46-36-23-104.k-telecom.org (46.36.23.104)
Host is up (0.71s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 100 IP addresses (6 hosts up) scanned in 16.91 seconds

```

En la captura se detecta un host con un puerto 80/tcp abierto.

- IP detectada: 52.202.2.227
- Puerto: 80/tcp open

Procedemos a escanearla usando nikto -h 52.202.2.227

```

- ***** RFIURL is not defined in nikto.conf--no RFI tests will run *****
- Nikto v2.5.0
-----
+ Target IP:      52.202.2.227
+ Target Hostname: 52.202.2.227
+ Target Port:    80
+ Start Time:     2025-05-17 22:46:55 (GMT-3)
-----
+ Server: awselb/2.0
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://52.202.2.227:443/
+ No CGI Directories found (use '-C all' to force check all possible dirs)

```

En la misma se detectan las siguientes vulnerabilidades:

Vulnerabilidad / Advertencia	Descripción
X-Frame-Options no presente	Podría permitir ataques de clickjacking, donde una página maliciosa embebe el sitio con elementos para hacer click y engañar al usuario.
X-Content-Type-Options no presente	Puede permitir ataques de tipo MIME sniffing, donde el navegador interpreta archivos con un tipo diferente al declarado, generando riesgos de ejecución de scripts maliciosos.
Redirección a HTTPS	La raíz / redirige automáticamente a https://52.202.2.227, lo cual es una buena práctica, pero el escaneo se limitó al puerto 80.
Sin directorios CGI detectados	No se hallaron scripts CGI clásicos, lo cual reduce la superficie de ataque tradicional.

4.4)

```
File Actions Edit View Help
kali@kali: ~
$ nmap -p 80 --open -Pn -T4 175.45.176.0/22 --max-retries 1 --min-rate 1000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-09 17:11 EDT
Nmap scan report for 175.45.176.68
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.69
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.71
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.75
Host is up (0.41s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.76
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.88
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.81
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.85
Host is up (0.43s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.176.91
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.177.1
Host is up (0.42s latency).
```

```
File Actions Edit View Help
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.177.1
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.177.10
Host is up (0.42s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for 175.45.177.11
Host is up (0.43s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1024 IP addresses (1024 hosts up) scanned in 29.04 seconds
kali@kali: ~
$
```

Resultados:

- Total de hosts con puerto 80 abierto: 12.
- Direcciones IP identificadas:

175.45.176.68, 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

Latencia promedio: ~0.42 segundos (indicador de conectividad estable).

Se identificaron **12 servidores web activos** en el rango IP asignado a Corea del Norte. La presencia de múltiples hosts con HTTP abierto sugiere una infraestructura centralizada, aunque sin datos adicionales no es posible determinar su propósito exacto.