

❖ WiFi Access Controller (Monitor de Red)

Una herramienta de monitoreo de red en tiempo real escrita en Python. Escanea la red local, identifica dispositivos mediante su dirección MAC, detecta intrusos y controla el límite de conexiones simultáneas por usuario.

Nota: Este proyecto utiliza una interfaz visual en terminal (TUI) gracias a la librería Rich.

🕒 Captura de Pantalla

Trabajo Final Estructura de dato > main.py > ...

```
15     class AccessController:
16
17         def escanear_red(self, rango_ip):
18             try:
19                 arp = ARP(pdst=rango_ip)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Users\Owner\Downloads\Tecnico Ciberseguridad\Estructura de Datos\Ciberseguridad> cd 'c:\Users\Owner\Downloads\Tecnico Ciberseguridad\Estructura de Datos\Ciberseguridad'; & 'c:\Users\Owner\anaconda3\python.exe' 'c:\Users\Owner\vscode\extensions\ms-python.debugpy-2025.16.0-win32-x64\bundled\libs\debugpy\launcher' '60106' '--' 'C:\Users\Owner\Downloads\Tecnico Ciberseguridad\Estructura de Datos\Ciberseguridad\Trabajo Final Estructura de dato\main.py'
```

Iniciando Monitor en 192.168.1.0/24...

Monitor WiFi (Activos: 10)

Usuario	IP	MAC	Estado
DESCONOCIDO	192.168.1.1	7C-39-53-BF-2E-4A	INTRUSO
DESCONOCIDO	192.168.1.8	C0-25-67-AD-DA-EC	INTRUSO
MiPc	192.168.1.10	70-CD-0D-54-6C-8F	OK
DESCONOCIDO	192.168.1.3	68-57-2D-D9-08-36	INTRUSO
DESCONOCIDO	192.168.1.4	18-69-D8-27-F9-7F	INTRUSO
DESCONOCIDO	192.168.1.6	5A-2B-33-76-26-27	INTRUSO
DESCONOCIDO	192.168.1.7	18-69-D8-29-4D-CF	INTRUSO
DESCONOCIDO	192.168.1.13	A4-CC-B3-8E-54-40	INTRUSO
EjemploUsuario	192.168.1.9	B0-73-9C-E7-4D-8C	OK
DESCONOCIDO	192.168.1.11	A4-CC-CE-3C-93-CE	INTRUSO

ALERTAS DE SEGURIDAD

```
[21:53:53] NO AUTORIZADO: MAC: 7C-39-53-BF-2E-4A en IP 192.168.1.1
[21:53:53] NO AUTORIZADO: MAC: C0-25-67-AD-DA-EC en IP 192.168.1.8
[21:53:53] NO AUTORIZADO: MAC: 68-57-2D-D9-08-36 en IP 192.168.1.3
[21:53:53] NO AUTORIZADO: MAC: 18-69-D8-27-F9-7F en IP 192.168.1.4
[21:53:53] NO AUTORIZADO: MAC: 5A-2B-33-76-26-27 en IP 192.168.1.6
[21:53:53] NO AUTORIZADO: MAC: 18-69-D8-29-4D-CF en IP 192.168.1.7
[21:53:53] NO AUTORIZADO: MAC: A4-CC-B3-8E-54-40 en IP 192.168.1.13
[21:53:53] NO AUTORIZADO: MAC: A4-CC-CE-3C-93-CE en IP 192.168.1.11
```

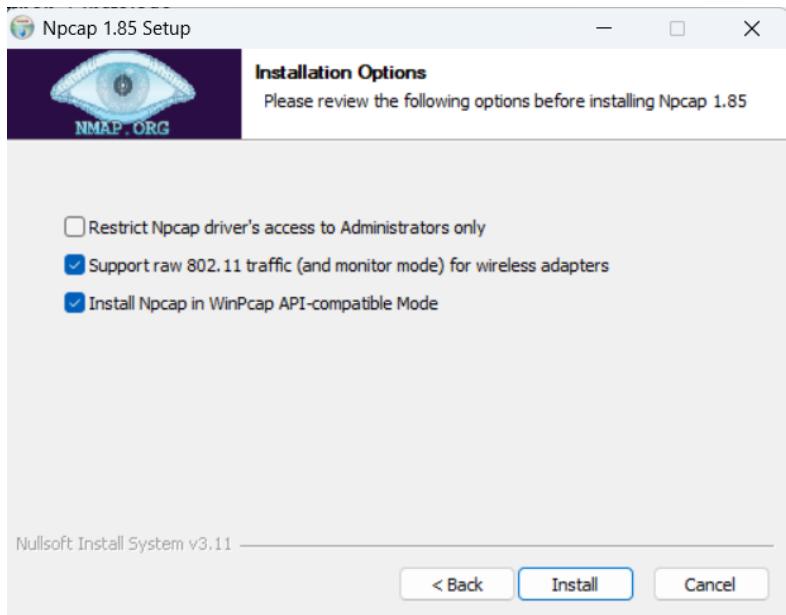
❖ Características Principales

- Escaneo en Tiempo Real:** Utiliza peticiones ARP para descubrir dispositivos en la red local.
- Detección de Intrusos:** Compara las MACs encontradas con una lista blanca (permitidos.txt). Si no está registrado, marca el dispositivo como **INTRUSO**.
- Control de Límite de Conexiones:** Permite definir un número máximo de dispositivos por usuario (ej. 1 por persona). Si un usuario conecta más dispositivos de los permitidos, genera una alerta de **LÍMITE EXCEDIDO**.
- Interfaz Visual Moderna:** Tabla de estados y panel de alertas en vivo.
- Agnóstico al formato MAC:** Lee direcciones MAC con guiones (-), dos puntos (:) o sin separadores.

❖ Requisitos Previos

Para Windows

1. Python 3 instalado.
2. **Importante:** Instalar [Npcap](#). Para que Python pueda "leer" el tráfico WiFi, necesitas este controlador especial, ya que Windows protege la tarjeta de red.
 - o Durante la instalación, asegúrate de marcar la casilla: "**Install Npcap in WinPcap API-compatible Mode**".



Para Linux / Mac

1. Python 3.
2. Permisos de `sudo` (necesarios para que Scapy acceda a la tarjeta de red).

❖ Instalación

1. Clona este repositorio:

Bash

```
https://github.com/The-Eliezer/Ciberseguridad.git
```

2. Instala las dependencias:

Bash

```
pip install -r requerimiento.txt
```

⚙️ Configuración

1. Definir Dispositivos Permitidos

Edita el archivo `permitidos.txt`. Agrega las direcciones MAC y el nombre del propietario. El formato es `MAC, Nombre`.

Ejemplo:

Plaintext

```
A4-B1-C2-D3-E4-F5, Administrador  
00:11:22:33:44:55, Invitado  
AA BB CC DD EE FF, TV Sala
```

2. Configurar la Red

Abre el archivo `main.py` y busca la línea `RANGO_RED`. Ajústala según tu IP local:

Python

```
# Si tu IP es 192.168.1.5  
RANGO_RED = "192.168.1.0/24"  
  
# Si tu IP es 192.168.0.14  
RANGO_RED = "192.168.0.0/24"
```

También puedes ajustar el límite de conexiones al iniciar el controlador:

Python

```
controlador = AccessController(limite_conexiones=1)
```

🔧 Uso

El script requiere permisos de administrador para enviar paquetes ARP.

En Windows (CMD / PowerShell como Administrador):

Bash

```
python main.py
```

En Linux / Mac:

Bash

```
sudo python main.py
```

⚠️ Descargo de Responsabilidad

Esta herramienta fue creada con fines educativos y de aprendizaje sobre estructuras de datos (Matrices, Vectores) y redes. Úsala solo en redes que te pertenezcan o donde tengas permiso para monitorear.