**Evaluación Contínua 2 - Hacking con Python**

**Chequeado**

Alumno: Gonzalo Benito

Email: [gu4n4rt@gmail.com](mailto:gu4n4rt@gmail.com)

…..............................................................................................................................................................

**Funcionamiento:**

La interfaz le muestra al usuario las interfaces de red que dispone. Se elige una, y comienza el escaneo a toda la red en la que se encuentra, mediante el análisis de su dirección y máscara de red.

Una vez identificada la red, en enviarán paquetes ARP con Scapy, para descubrir todos los hosts vivos, almacenando y analizando las respuestas, para crear instancias de la clase “cliente” con la información devuelta.

Posteriormente, realiza un escaneo de puertos (TCP Stealth y UDP) a los hosts disponibles, también con Scapy, y almacena los resultados en listas atributo de los objetos de la clase “cliente”.

Realiza banner grabbing a todos los puertos abiertos mediante la librería Socket y extrae los banners.

Por último, exporta los resultados a un fichero JSON y lo envía a un servidor web mediante una petición POST requests.

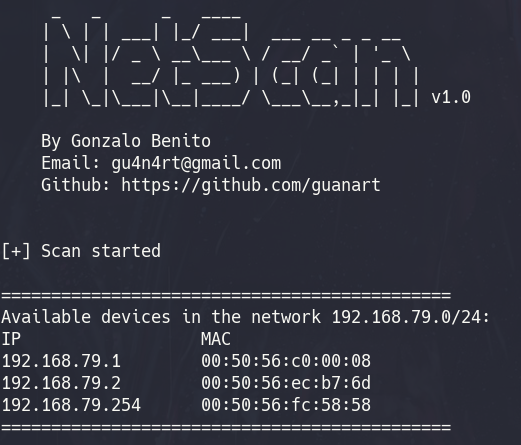
**Descarga**

*git clone* [*https://github.com/Guanart/NetScan*](https://github.com/Guanart/NetScan)

*pip install -r requirements*

**Imágenes**

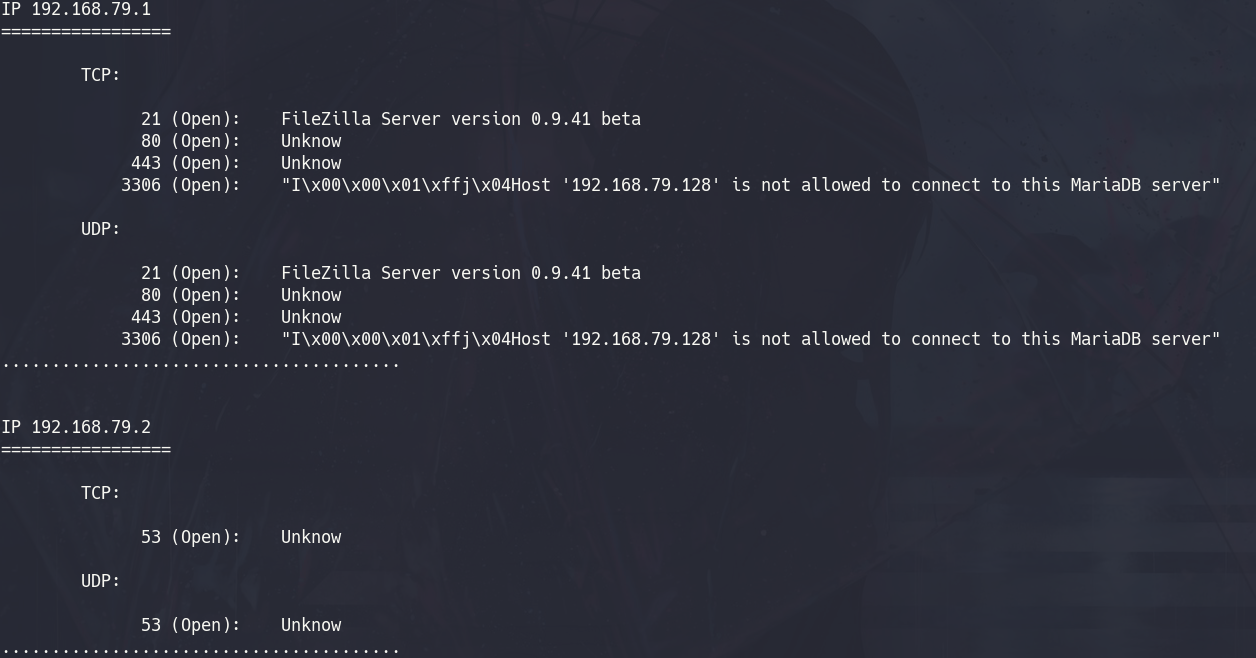
Ejecución normal indicando nuestra interfaz de red:



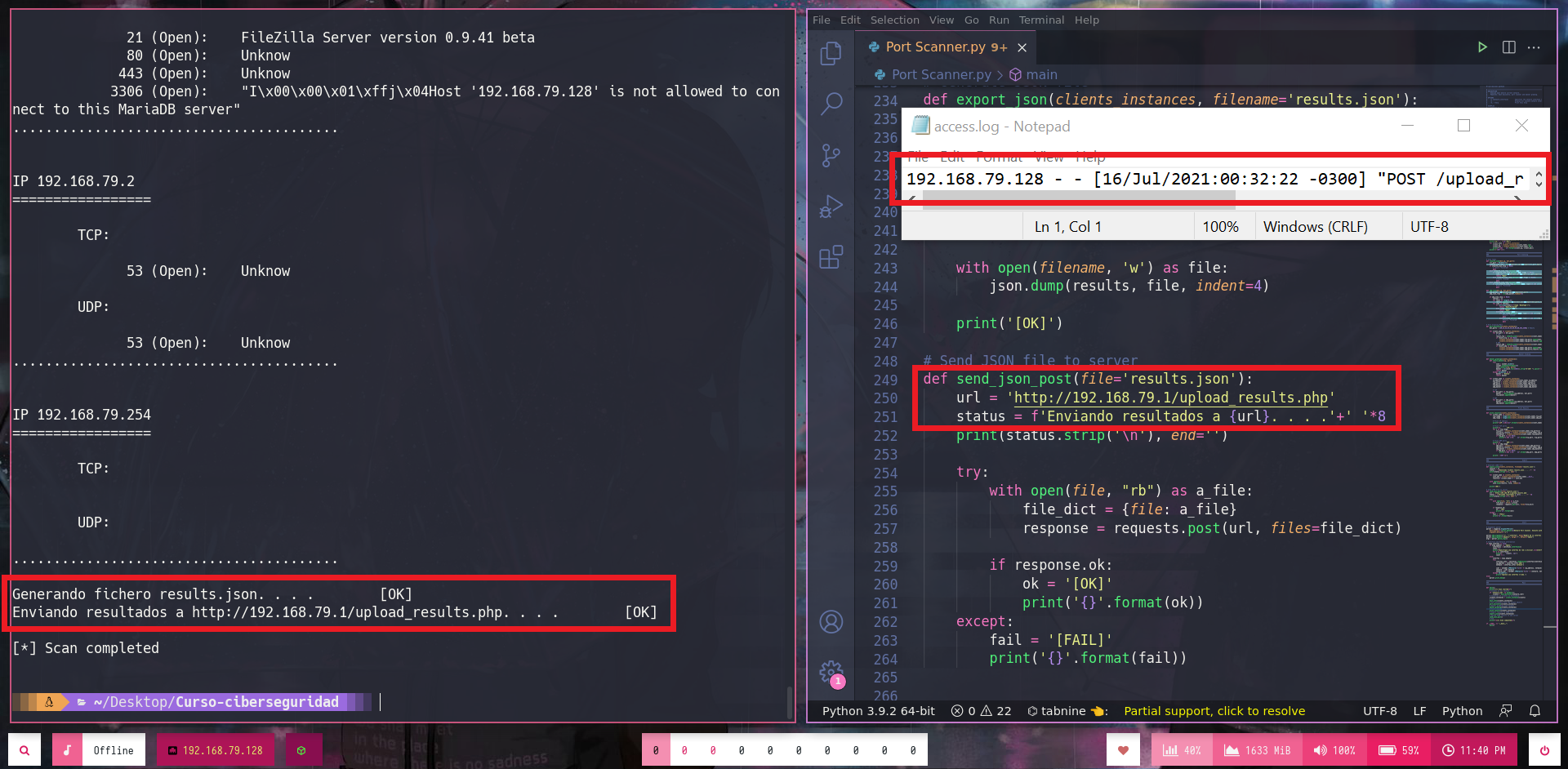
Si no le indicamos la interfaz, nos mostrará las que tenemos disponibles:



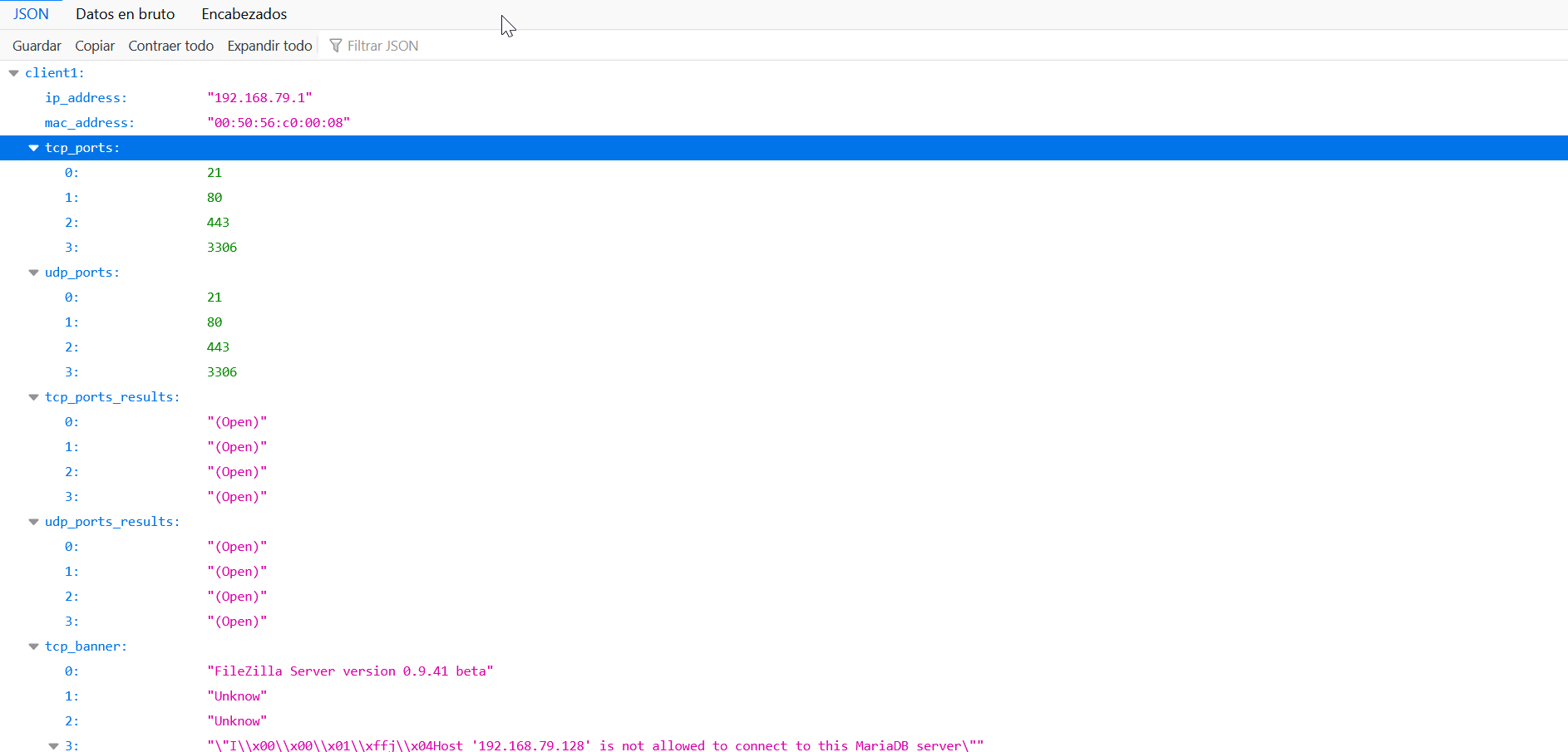
Resultados:



Los resultados se exportan a un fichero JSON y se envía a un servidor web que indiquemos:



Para poder almacenar el fichero en el servidor, debemos contar con el código backend (como PHP) necesario:



***Prueba en Android-Termux (Kali Nethunter)***

Probé el script en Kali Nethunter. Detecto satisfactoriamente las interfaces de red. El inconveniente surge con la librería Scapy, que requiere de permisos de superusuario.

