

Consultar Fabricantes de tarjetas de red a través de una API

Stefanny Montero Vasquez, stefanny.montero@alumnos.uv.cl

Yoselin Cornejo Rivera, yoselin.cornejo@alumnos.uv.cl

Introducción

La tarea consistió en desarrollar una herramienta en línea de comandos llamada "OUILookup" utilizando un programa de código en Python, que permitió consultar el fabricante de una tarjeta de red a partir de su dirección MAC. Se utilizó una API REST pública para obtener la información, implementando funcionalidades tanto para sistemas Windows como Linux. El proyecto incluyó documentación, pruebas y fue alojado en un repositorio GitHub para su revisión.

1. Descripción del problema

El problema que se resuelve es la identificación del fabricante de un dispositivo a partir de su dirección MAC, dado que esta información no es visible directamente. Para solucionarlo, se creó una herramienta de línea de comandos en Python llamada OUILookup.

Diseño de la solución:

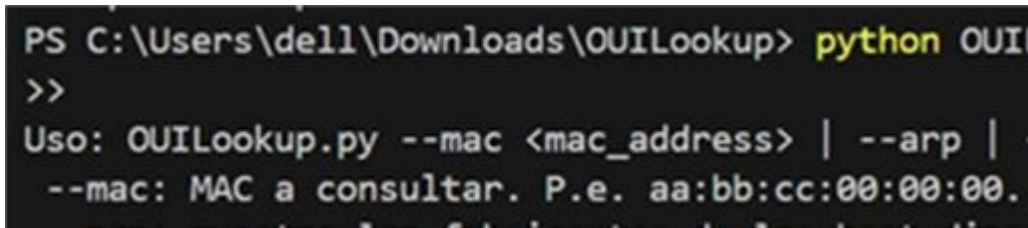
Consulta de fabricantes: A través de la función `get_mac_info(mac_address)`, se utiliza una API pública para obtener el fabricante de una dirección MAC.

Tabla ARP: Con `show_arp_table()`, el programa extrae la tabla ARP del sistema local (Windows o Linux), consulta los fabricantes de las MAC presentes y los muestra.

Parámetros de uso: El programa acepta parámetros como `--help`, `--mac` y `--arp` para mostrar ayuda, consultar un fabricante específico o listar los fabricantes de la tabla ARP.

2. Ejemplos de uso:

2.1 Parametro --help

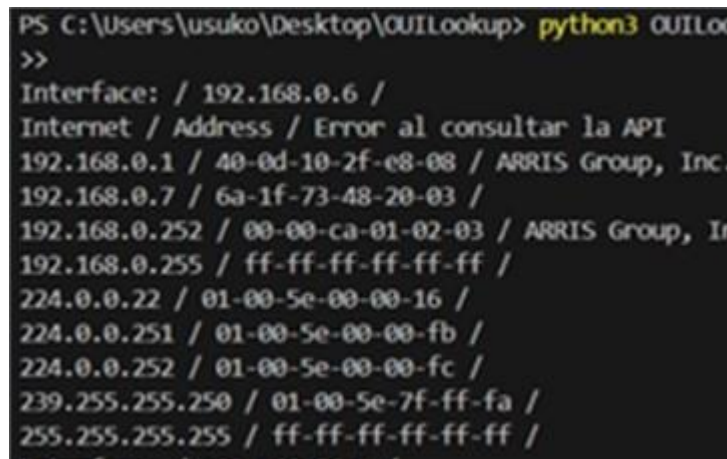


```
PS C:\Users\dell\Downloads\OUILookup> python OUILookup.py --help
>>
Uso: OUILookup.py --mac <mac_address> | --arp | --help
--mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
```

Imagen 1

En la imagen 1 la salida muestra las opciones disponibles para ejecutar el programa OUILookup.py. Entre las opciones están consultar un fabricante de MAC (--mac), listar los fabricantes de las direcciones MAC presentes en la tabla ARP (--arp), y mostrar este mensaje de ayuda (--help).

2.2 Parametro --arp

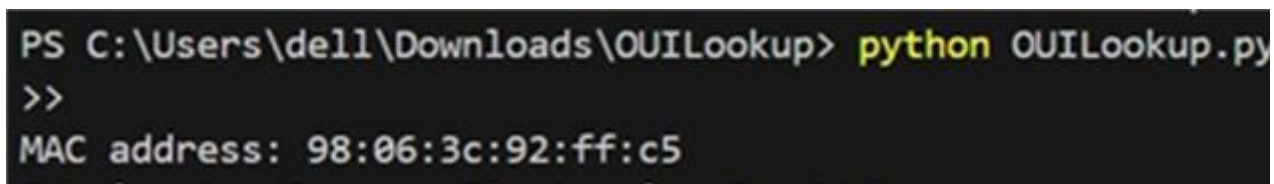


```
PS C:\Users\usuko\Desktop\OUILookup> python3 OUILookup.py --arp
>>
Interface: / 192.168.0.6 /
Internet / Address / Error al consultar la API
192.168.0.1 / 40-0d-10-2f-e8-08 / ARRIS Group, Inc.
192.168.0.7 / 6a-1f-73-48-20-03 /
192.168.0.252 / 00-00-ca-01-02-03 / ARRIS Group, Inc.
192.168.0.255 / ff-ff-ff-ff-ff-ff /
224.0.0.22 / 01-00-5e-00-00-16 /
224.0.0.251 / 01-00-5e-00-00-fb /
224.0.0.252 / 01-00-5e-00-00-fc /
239.255.255.250 / 01-00-5e-7f-ff-fa /
255.255.255.255 / ff-ff-ff-ff-ff-ff /
```

Imagen 2

En la imagen 2 la salida muestra la lista de direcciones MAC disponibles en la tabla ARP del sistema y los fabricantes asociados a ellas, en este caso, el programa consulta las direcciones MAC detectadas localmente y devuelve los fabricantes, como ARRIS Group, Inc..

2.3 Parametro --mac



```
PS C:\Users\dell\Downloads\OUILookup> python OUILookup.py --mac 98:06:3c:92:ff:c5
>>
MAC address: 98:06:3c:92:ff:c5
```

Imagen 3

En la imagen 3 se muestra el fabricante de la dirección MAC específica y el tiempo de respuesta.

3. Casos de prueba



Imagen 4

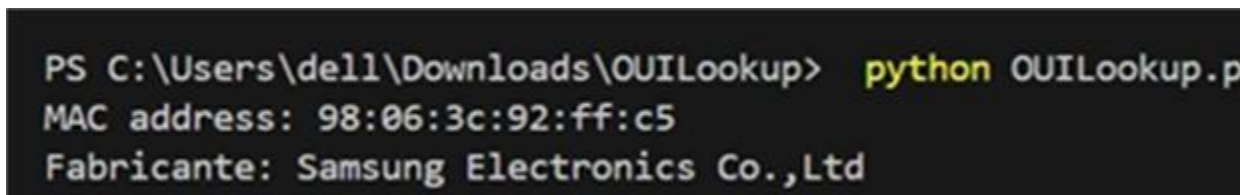


Imagen 5

Las imágenes 4 y 5 muestran el resultado de consultar la dirección MAC 98:06:3c:92:ff:c5 con el programa OUILookup. La imagen 4 presenta detalles obtenidos de la API, que identifican al fabricante como Samsung Electronics Co.,Ltd, junto con su ubicación y detalles del bloque MAC, la imagen 5 muestra la ejecución en la terminal, donde se confirma el fabricante y se reporta un tiempo de respuesta de 776.22 ms.



Imagen 6

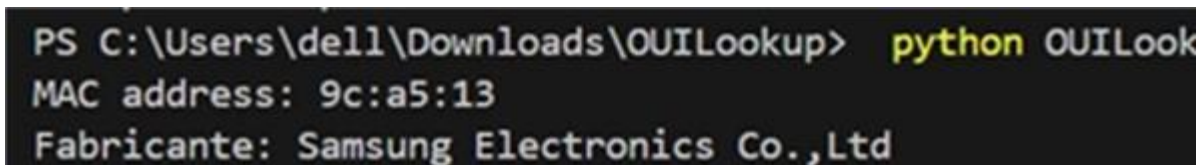


Imagen 7

En las imágenes 6 y 7 se muestran la consulta de la dirección MAC 9c:a5:13 usando el programa OUILookup. En la imagen 6 se detalla la información obtenida de la API, que identifica al fabricante como Samsung Electronics Co.,Ltd, junto con su dirección en Corea del Sur, en la imagen 7 muestra la ejecución en la terminal, confirmando al fabricante y mostrando un tiempo de respuesta de 753.09 ms.

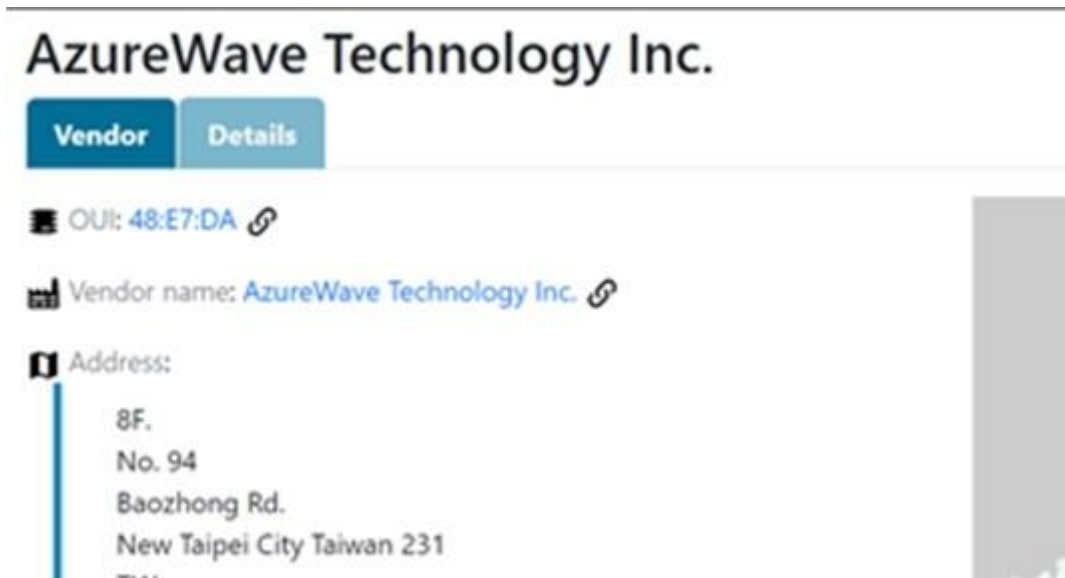


Imagen 8

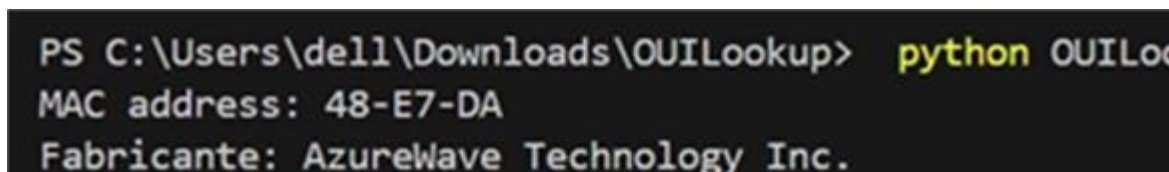


Imagen 9

Las imágenes 8 y 9 muestran la consulta de la dirección MAC 48-E7-DA usando el programa OUILookup. En la imagen 8 presenta los detalles obtenidos de la API, identificando al fabricante como AzureWave Technology Inc. con su dirección en Taiwán, en la imagen 9 se muestra la ejecución en la terminal, confirmando el fabricante y mostrando un tiempo de respuesta de 1626.36 ms.

4. Implementación

En esta sección, se explicará el código implementado. El código se centra en la creación de una herramienta llamada "OUILookup", la cual permite realizar consultas sobre fabricantes de tarjetas de red a partir de direcciones MAC utilizando una API REST. El programa está estructurado en diferentes funciones clave.

Función get_mac_info: Esta función realiza una consulta a la API de maclookup.app para obtener el fabricante de la dirección MAC ingresada. Se maneja cualquier error de conexión mediante un bloque try-except que asegura que el programa no se detenga inesperadamente ante fallos en la red o en la API.

Función show_arp_table: Esta función detecta el sistema operativo (Windows o Linux) y ejecuta el comando adecuado para obtener la tabla ARP. A partir de esta tabla, extrae las direcciones MAC y consulta el fabricante correspondiente utilizando la función

get_mac_info. La salida muestra la IP, la dirección MAC y el fabricante, todo en una sola línea.

Función main: La función principal utiliza la librería getopt para procesar los argumentos de la línea de comandos. Según el argumento recibido (--mac o --arp), realiza la consulta de una MAC específica o muestra la tabla ARP. El tiempo de respuesta se mide para las consultas de MAC, lo que proporciona información adicional sobre el rendimiento de la API.

Los principales desafíos que se encontraron en esta fase fue el manejo de excepciones al manejar posibles errores de conexión con la API, lo cual fue resuelto implementando un bloque try-except para garantizar la robustez del programa y la compatibilidad del comando ARP, dado que los sistemas Windows y Linux utilizan diferentes comandos para acceder a la tabla ARP, se implementó una detección del sistema operativo mediante la función platform.system() para ejecutar el comando adecuado en cada caso.

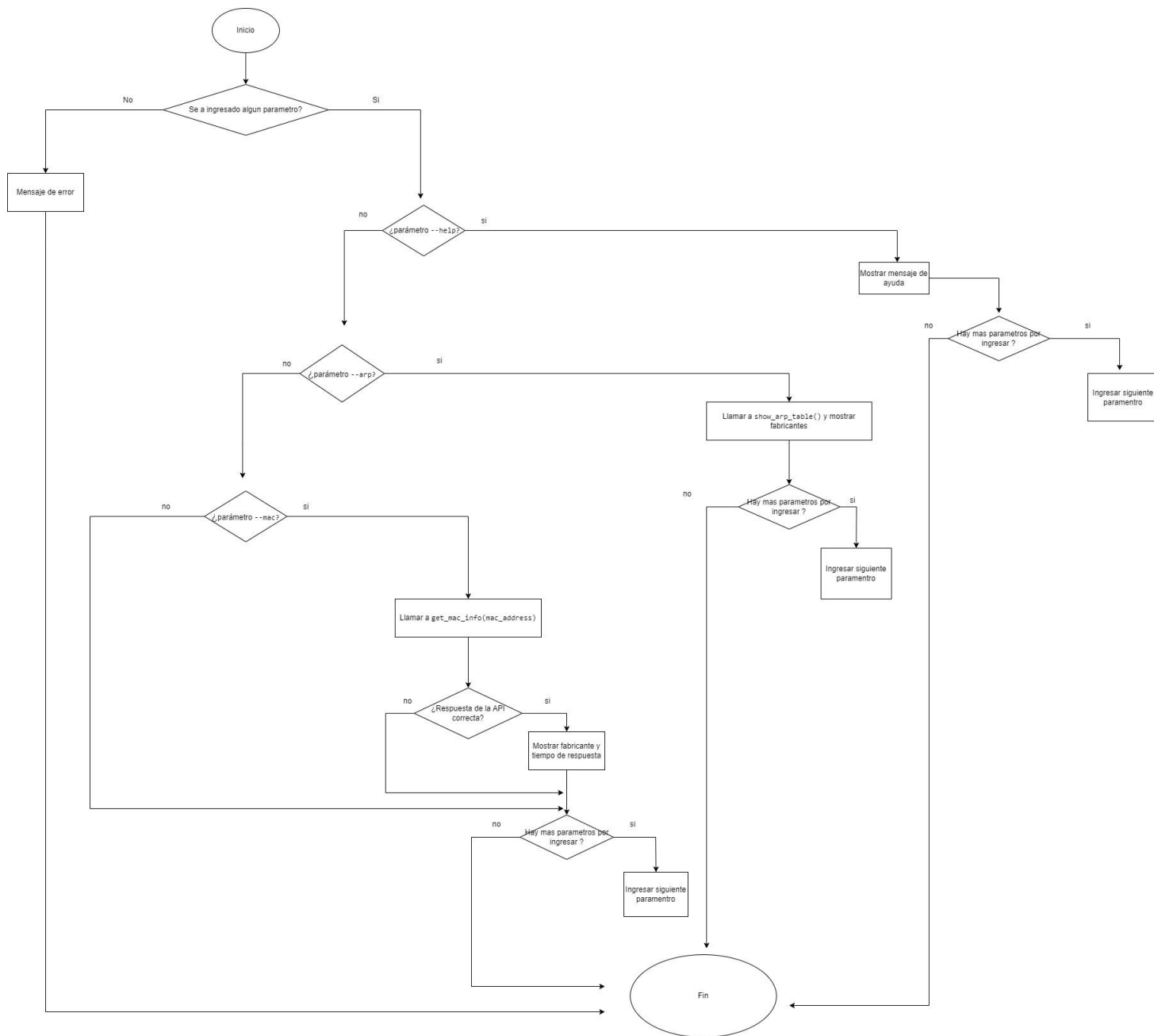
5. Mac aleatorias

La aleatorización de direcciones MAC es una técnica implementada en muchos dispositivos electrónicos modernos, como smartphones y laptops, para mejorar la privacidad del usuario. Su función es generar una dirección MAC temporal y aleatoria en lugar de utilizar la dirección MAC permanente del dispositivo. Esto se usa principalmente durante el proceso de descubrimiento de redes Wi-Fi, como al enviar solicitudes de sondeo para conectarse a puntos de acceso.

El propósito de esta técnica es dificultar el seguimiento de dispositivos a largo plazo, ya que la dirección MAC es un identificador único que puede ser capturado por terceros al monitorear redes públicas. Sin embargo, aunque la aleatorización ofrece un mayor nivel de privacidad, no es completamente infalible. ¹Existen vulnerabilidades conocidas, como la posibilidad de que el dispositivo vuelva a utilizar su dirección MAC original al conectarse a una red, lo que expone nuevamente su identidad.²

Además, algunos estudios han demostrado que es posible rastrear dispositivos incluso cuando se emplea la aleatorización de MAC, ya que algunos comportamientos específicos de los dispositivos, como las características de las solicitudes de sondeo, pueden permitir a atacantes identificar un dispositivo a pesar de que cambie su dirección MAC.

6. Diagrama de flujo



Este diagrama muestra el flujo del programa OUILookup. Dependiendo de los parámetros ingresados (--help, --arp, --mac), el programa muestra ayuda, lista los fabricantes en la tabla ARP o consulta el fabricante de una dirección MAC específica. También maneja errores cuando no se ingresan parámetros válidos.

8. Conclusión

La herramienta **OUILookup** desarrollada permite consultar de manera eficiente el fabricante de una tarjeta de red a partir de su dirección MAC mediante una API pública. Durante su implementación, se superaron desafíos como el manejo de errores de conexión y la compatibilidad con sistemas operativos diferentes. La solución es robusta, versátil y mejora el acceso a la información sobre dispositivos de red, optimizando el proceso de identificación de fabricantes desde la tabla ARP.

9. Referencias

[1] Vanhoef, M.; Piessens, F. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (ASIACCS) 2016, 13, 14-25.

<https://papers.mathyvanhoef.com/asiaccs2016.pdf>

[2] Vanhoef, M.; Matte, J.; Halderman, J.A. A Study of MAC Address Randomization in Mobile Devices and When it Fails. arXiv 2017, 1703.02874, 1-17.

<https://arxiv.labs.arxiv.org/html/1703.02874>