

Tarea 2 Redes de Computadores

Diego Peña Gutiérrez, diego.penag@alumnos.uv.cl

1. Introducción

En este informe se presentará el desarrollo de una herramienta llamada “OUILookup”, diseñada para consultar el fabricante de una tarjeta de red a partir de su dirección MAC. La importancia de esta herramienta radica en su capacidad de facilitar la identificación de dispositivos en una red, lo que puede ser útil en tareas de redes y seguridad informática. El objetivo principal del trabajo es implementar una solución fundamental que utilice una API REST pública para obtener información sobre los fabricantes de direcciones MAC.

2. Descripción del problema y diseño de la solución

El problema para resolver es el como identificar el fabricante de dispositivos en una red a partir de sus direcciones MAC. Se requería una solución que fuera accesible y fácil de usar por lo cual se decidió crear un código en Python, el cual usa una arquitectura cliente-servidor donde la herramienta actúa como cliente que realiza peticiones a la API de *maclookup.app*.

3. Implementación

El código de “OUILookup” se compone de varias funciones principales. La función `obtener_detalle_mac(mac)` se encarga de realizar una consulta a la API y devuelve el fabricante correspondiente. La función `mostrar_detalle_mac(mac)` gestiona la presentación de la información mostrando la dirección MAC consultada, el fabricante y el tiempo de respuesta. Durante la implementación se enfrentaron desafíos relacionados con la gestión de errores de la API que se resolvían mediante manejo de excepciones.

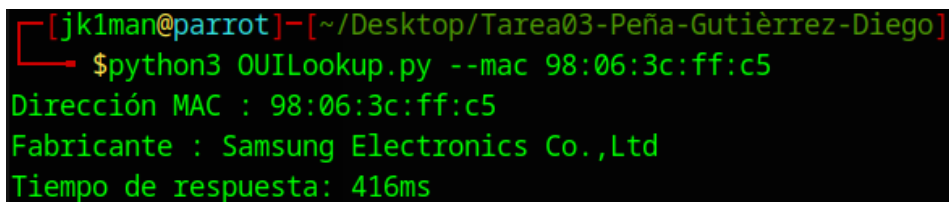
4. Pruebas

Las pruebas se realizaron usando direcciones MAC conocidas y no conocidas para evaluar la efectividad de la herramienta. A continuación, se presentan algunos ejemplos de casos de prueba:

- Caso de prueba con MAC válida

Entrada `python3 OUILookup.py --mac 98:06:3c:ff:c5`

Salida:



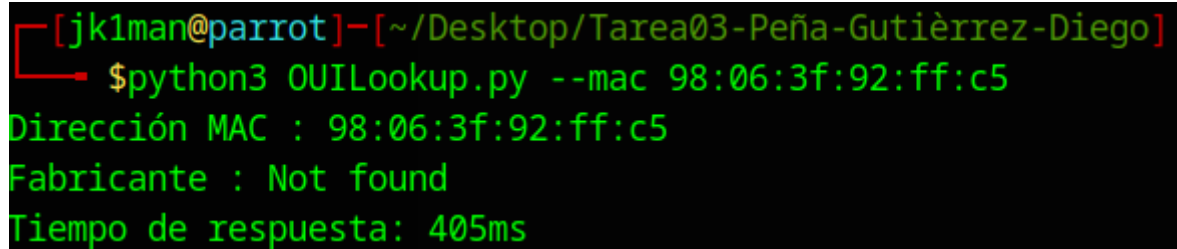
```
[jk1man@parrot] - [~/Desktop/Tarea03-Peña-Gutiérrez-Diego]
$python3 OUILookup.py --mac 98:06:3c:ff:c5
Dirección MAC : 98:06:3c:ff:c5
Fabricante : Samsung Electronics Co.,Ltd
Tiempo de respuesta: 416ms
```

Figura 1: MAC válida

- **Caso prueba con una MAC no válida**

Entrada `python3 OUILookup.py --mac 98:06:3f:92:ff:c5`

Salida:



```
[jk1man@parrot]--[~/Desktop/Tarea03-Peña-Gutiérrez-Diego]  
$python3 OUILookup.py --mac 98:06:3f:92:ff:c5  
Dirección MAC : 98:06:3f:92:ff:c5  
Fabricante : Not found  
Tiempo de respuesta: 405ms
```

Figura 2 MAC no válida

5. Discusión y conclusiones

Se obtuvo una herramienta funcional que cumple con los requisitos establecidos. Durante el proceso de desarrollo se aprendió sobre el manejo de APIs y la importancia de gestionar adecuadamente los errores. Aunque la implementación se podrían agregar mejoras como un sistema de caché para almacenar resultados MACs consultados frecuentemente, lo que podría reducir el tiempo de respuesta.

6. Referencias

[1] maclookup.app. "API Documentation" 2024 <https://maclookup.app>.

[2] Oficial Python Documentation "getopt -C-style parser for command line options." <https://docs.python.org/3/library/getopt.html>

7. Diagrama de flujo

