

Tarea 2 – OUILookup.

Juan Barraza, juan.barraza@alumnos.uv.cl

Walter Castillo, walter.castillo@alumnos.uv.cl

1. Introducción

Este informe describe el desarrollo de una herramienta en línea de comandos llamada OUILookup implementada en Python. El propósito de esta herramienta es consultar el fabricante de una tarjeta de red a partir de su dirección MAC, utilizando una API REST pública. Este tipo de herramienta es útil en redes de computadores, donde se pueden identificar dispositivos por sus direcciones MAC para tareas de administración y seguridad. El objetivo principal es implementar una solución funcional que permita obtener información sobre el fabricante de una tarjeta de red, tanto a partir de una MAC específica como de las MAC presentes en la tabla ARP del sistema.

2. Descripción del problema y diseño de la solución

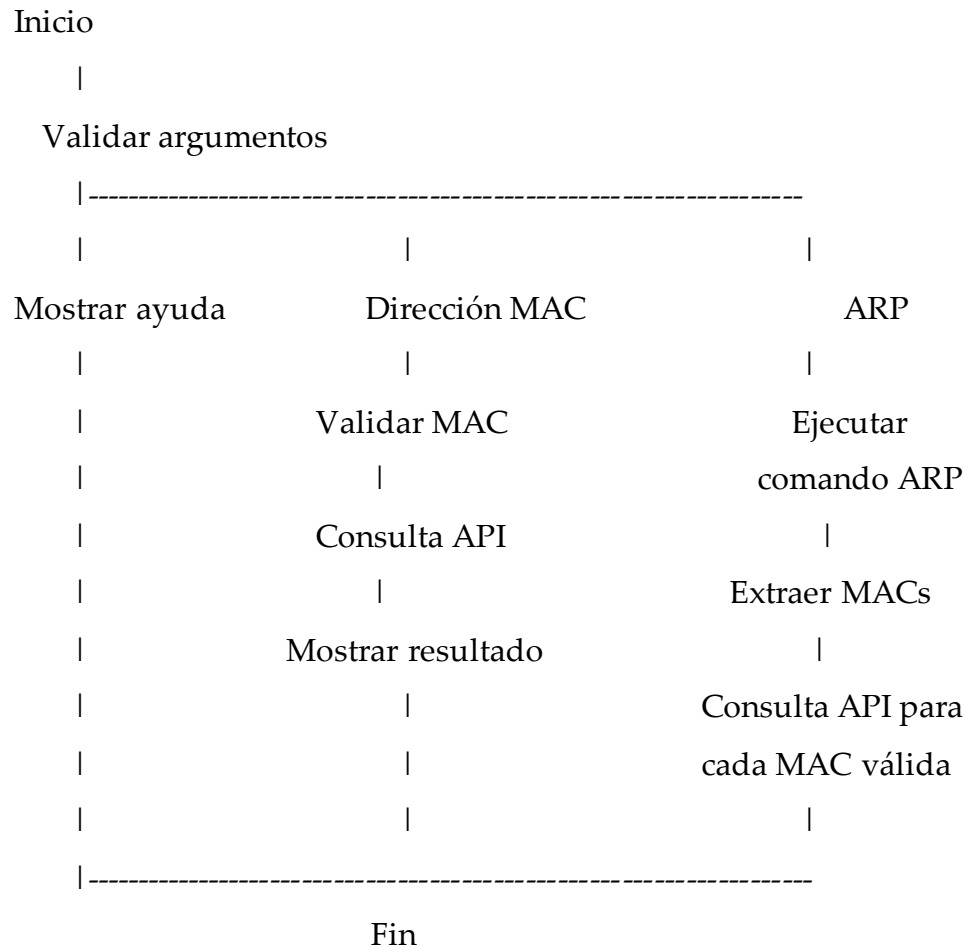
El problema a resolver consiste en crear un programa que permita consultar el fabricante de una tarjeta de red a partir de su dirección MAC. Para ello, se requiere de una API REST que proporcione la base de datos de fabricantes de dispositivos. Los requerimientos principales incluyen:

- Procesamiento de parámetros de línea de comandos.
- Consulta de la dirección MAC mediante la API `maclookup.app`.
- Consulta de fabricantes de las MAC presentes en la tabla ARP.

La solución se diseñó en Python siguiendo un modelo sencillo de arquitectura que permite:

- Ejecutar comandos de terminal para obtener direcciones MAC.
- Utilizar la librería `requests` para interactuar con la API REST.

A continuación se presenta un diagrama de flujo simplificado del proceso general:



3. Implementación

En esta sección, debe explicar las partes más importantes del código, incluyendo cómo se implementaron las clases. En algunas ocasiones será necesario utilizar diagramas de flujo para explicar el funcionamiento de las clases. También debe describir los principales desafíos que se encontraron en esta fase y cómo fueron resueltos.

La implementación se llevó a cabo en Python utilizando las siguientes herramientas y bibliotecas:

- **getopt** para el manejo de argumentos en la línea de comandos.
- **requests** para interactuar con la API REST pública.

La estructura principal del código se basa en dos funcionalidades:

1. **Consulta de una MAC específica:** El usuario puede ingresar una dirección MAC y el programa realiza una solicitud a la API para obtener el fabricante.
2. **Consulta de MACs en la tabla ARP:** El programa ejecuta el comando ARP en el sistema y extrae las direcciones MAC disponibles, consultando los fabricantes correspondientes.

Desafíos importantes:

- Gestionar los errores de conexión a la API y asegurar la compatibilidad con Windows y Linux.
- Extraer las direcciones MAC de la tabla ARP de forma eficiente mediante expresiones regulares.

4. Pruebas

Explicar cómo se realizaron las pruebas del código. Se deben incluir ejemplo de los casos de pruebas, sus resultados y cómo se aseguró el funcionamiento correcto del código.

Se realizaron diversas pruebas para asegurar el correcto funcionamiento del programa:

Prueba con una MAC conocida:

```
tarea2-OUILookup>python OUILookup.py --mac 98:06:3c:92:ff:c5
```

Resultado esperado:

```
MAC address: 98:06:3c:92:ff:c5  
Fabricante: Samsung Electronics Co.,Ltd  
Tiempo de respuesta: 717ms
```

Prueba con una MAC inexistente:

```
>python OUILookup.py --mac 00:00:00:00:00:00
```

Resultado esperado:

```
MAC address: 00:00:00:00:00:00  
Fabricante: NOT FOUND  
Tiempo de respuesta: 0ms
```

Prueba de ARP:

```
>python OUILookup.py --arp
```

Resultado esperado:

```
MAC address: 18-35-d1-21-f8-b0  
Fabricante: ARRIS Group, Inc.  
Tiempo de respuesta: 710ms  
MAC address: 50-0f-f5-e0-97-e0  
Fabricante: Tenda Technology Co.,Ltd.Dongguan branch  
Tiempo de respuesta: 630ms  
MAC address: 5c-93-a2-d1-2f-d8  
Fabricante: Liteon Technology Corporation  
Tiempo de respuesta: 416ms  
MAC address: 00-00-ca-01-02-03  
Fabricante: ARRIS Group, Inc.  
Tiempo de respuesta: 648ms  
MAC address: ff-ff-ff-ff-ff-ff  
Fabricante: NOT FOUND  
Tiempo de respuesta: 632ms
```

5. Funcionamiento de las Direcciones MAC Aleatorias en Dispositivos Electrónicos

Las direcciones MAC (Media Access Control) son identificadores únicos asignados a las interfaces de red en dispositivos electrónicos, fundamentales para su identificación en redes. En el contexto de la privacidad, las direcciones MAC aleatorias se han vuelto una herramienta importante para proteger la identidad de los usuarios, especialmente en redes Wi-Fi públicas.

5.1. Concepto de Direcciones MAC Aleatorias

Las direcciones MAC aleatorias son direcciones generadas de forma temporal y aleatoria por los dispositivos para ocultar su verdadera dirección MAC. Este enfoque es especialmente útil para proteger la privacidad del usuario en entornos donde pueden estar expuestos a la vigilancia o al seguimiento no deseado.

5.2. Cómo Funciona

- **Generación de Direcciones Aleatorias:** Los dispositivos generan una nueva dirección MAC aleatoria al conectarse a una red, utilizando una dirección diferente cada vez. Esta dirección se compone de 48 bits, representados en 12 dígitos hexadecimales (por ejemplo, 01:23:45:67:89:AB).
- **Uso Temporal:** La dirección MAC aleatoria se utiliza únicamente mientras el dispositivo está conectado a la red. Al desconectarse, puede generar una nueva dirección la próxima vez que se conecte, dificultando así el seguimiento del dispositivo.
- **Activación de la Funcionalidad:** La opción de usar direcciones MAC aleatorias suele estar habilitada en las configuraciones de privacidad de los dispositivos, como smartphones y laptops. Tanto **iOS** como **Android** han integrado esta funcionalidad en sus sistemas operativos.

5.3. Beneficios de las Direcciones MAC Aleatorias

- **Privacidad Mejorada:** Esta práctica mejora la privacidad del usuario al dificultar que terceros, como anunciantes o entidades gubernamentales, sigan a los usuarios a través de diferentes redes.
- **Seguridad:** Cambiar frecuentemente la dirección MAC reduce el riesgo de ataques de suplantación de identidad (spoofing) y complica la creación de perfiles de comportamiento basados en la dirección MAC.

5.4. Limitaciones

- **Problemas de Conectividad:** Algunos sistemas de autenticación que dependen de direcciones MAC específicas pueden no funcionar correctamente con direcciones aleatorias, lo que podría afectar la conectividad en ciertas redes.
- **Compatibilidad:** No todos los dispositivos o redes están configurados para gestionar direcciones MAC aleatorias, lo que puede generar problemas en redes más antiguas o configuraciones específicas.

6. Discusión y conclusiones

El desarrollo de la herramienta OUILookup permitió poner en práctica conceptos clave de redes de computadores, como la identificación de dispositivos a través de direcciones MAC y la interacción con API REST. El programa cumple con los requisitos planteados y se mostró eficiente en la consulta de fabricantes, tanto para una MAC específica como para las direcciones disponibles en la tabla ARP.

Posibles mejoras:

Ampliar el soporte para diferentes APIs en caso de que la API utilizada quede obsoleta.

Mejorar la interfaz de usuario para ofrecer resultados más claros y detallados.

Implementar un sistema de caché para acelerar futuras consultas de MACs previamente procesadas.

7. Referencias

[1] MACLookup API: <https://maclookup.app>

[2] Documentación de Python: <https://docs.python.org/3/library/getopt.html>

[3] Apple. (n.d.). Use private Wi-Fi addresses on your iPhone, iPad, or iPod touch. Recuperado de Apple Support.

[4] Android Developers. (n.d.). Wi-Fi Privacy. Recuperado de Android Developer Documentation.