

Tarea 2: Consultar Fabricantes de tarjetas de red a través de una API

Lucas Rojas Reyes, lucas.rojas@alumnos.uv.cl

Christopher O'Kinggton, christopher.okinggton@alumnos.uv.cl

1. Introducción

Dentro del ámbito de redes de computadoras la consulta y gestión de información relacionada con las interfaces de red es fundamental para la administración y seguridad de los sistemas informáticos. Este crucial proceso será vital para poder hacer uso de datos relevantes sobre tarjetas de red y sus fabricantes. El conocimiento adquirido será escalable a múltiples usos que sean externos a los solicitados y plantean el uso de conocimientos previamente adquiridos, esto debido a la común necesidad de obtener información detallada sobre los dispositivos conectados.

El objetivo de este trabajo es desarrollar una herramienta de consulta de información por medio de una API REST, la cual posteriormente será analizada y usada para investigar el funcionamiento de las direcciones MAC aleatorias en dispositivos electrónicos.

2. Descripción del problema y diseño de la solución

Esta tarea tiene como propósito desarrollar una herramienta basada en línea de comandos llamada "OUILookup" en Python, el programa hará uso de una API REST pública y permitirá consultar a fabricantes de tarjetas de red a través de su dirección MAC.

La herramienta de línea de comandos "OUILookup" deberá ser capaz de consultar el fabricante de una tarjeta de red a través de su dirección MAC usando la API REST pública "<https://maclookup.app>" y mostrar la tabla ARP con las MAC y fabricantes.

Al desarrollar nuestra solución deberemos considerar múltiples casos de uso, estos incluyen:

--mac: MAC a consultar.

--arp: muestra los fabricantes de los host disponibles en la tabla arp.

--help: muestra este mensaje y termina.

Adicionalmente el uso sin parámetros deberá tener la misma funcionalidad que "--help", un ejemplo de la salida esperada es:

```
$ python3 OUILookup.py --mac 98:06:3c:92:ff:c5
MAC address : 98:06:3c:92:ff:c5
Fabricante   : Samsung Electronics Co.,Ltd
Tiempo de respuesta: 17ms
```

Figura 1. Salida esperada usando Windows.

3. Implementación de OUILookup

Se hace de Python y la API REST publica para crear la herramienta de consulta por la línea de comandos, esta analiza los argumentos recibidos (`--mac`, `--arp` o `--help`) y posteriormente procesa la solicitud correspondiente mediante dos funciones principales.

Al usar `--mac` y obtener la dirección MAC haremos una solicitud mediante el API REST y devolveremos el nombre del fabricante y tiempo de respuesta.

De la misma forma al usar `--arp` obtendremos la tabla asociada a la conexión local.

Podemos ver el comportamiento descrito en el siguiente diagrama:



Figura 2. Diagrama de flujo del código.

4. Pruebas

Con el objetivo de probar el funcionamiento del código desarrollado se ingresarán las MACs de entrada solicitadas en el apartado de casos de prueba de la pauta y otros parámetros.

4.1. Pruebas con parámetro `--mac`

Para comprobar un fabricante específico utilizaremos los siguientes comandos:

1. 98:06:3c:92:ff:c5

```
PS C:\Andor> python OUILookup.py --mac 98:06:3c:92:ff:c5
MAC address: 98:06:3c:92:ff:c5
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 942.96 ms
```

Figura 3. Prueba de código creado N° 1.

2. 9c:a5:13

```
PS C:\Andor> python OUILookup.py --mac 9c:a5:13
MAC address: 9c:a5:13
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 628.12 ms
```

Figura 4. Prueba de código creado N° 2.

3. 48-E7-DA

```
PS C:\Andor> python OUILookup.py --mac 48-E7-DA
MAC address: 48-E7-DA
Fabricante: AzureWave Technology Inc.
Tiempo de respuesta: 942.64 ms
```

Figura 5. Prueba de código creado N° 3.

4.2. Pruebas con parámetro `--arp`

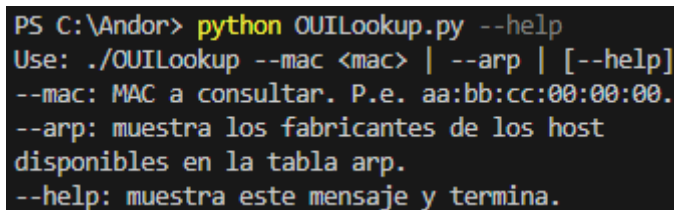
Para obtener la tabla ARP podemos usar el siguiente comando:

```
PS C:\Andor> python OUILookup.py --arp
IP: 192.168.1.1 / MAC: 88:de:7c:fd:27:b0 / Fabricante: ('ASKEY COMPUTER CORP', 933.9303970336914)
IP: 192.168.1.85 / MAC: 64:6e:69:62:06:f3 / Fabricante: ('Liteon Technology Corporation', 938.8821125030518)
IP: 192.168.1.89 / MAC: e8:d8:d1:28:a4:1f / Fabricante: ('HP Inc.', 1855.3211688995361)
IP: 192.168.1.90 / MAC: 60:f2:62:67:2b:ef / Fabricante: ('Intel Corporate', 917.2689914703369)
IP: 192.168.1.220 / MAC: 50:a5:dc:e1:6c:a9 / Fabricante: ('ARRIS Group, Inc.', 956.2642574310303)
IP: 192.168.1.224 / MAC: a4:98:13:72:b4:85 / Fabricante: ('ARRIS Group, Inc.', 828.5422325134277)
IP: 255.255.255.255 / MAC: ff:ff:ff:ff:ff:ff / Fabricante: ('', 834.8691463470459)
IP: 224.0.0.2 / MAC: 01:00:5e:00:00:02 / Fabricante: ('', 946.4423656463623)
IP: 224.0.0.22 / MAC: 01:00:5e:00:00:16 / Fabricante: ('', 785.008430480957)
IP: 224.0.0.251 / MAC: 01:00:5e:00:00:fb / Fabricante: ('', 943.3543682098389)
IP: 224.0.0.252 / MAC: 01:00:5e:00:00:fc / Fabricante: ('', 627.7158260345459)
IP: 239.255.255.250 / MAC: 01:00:5e:7f:ff:fa / Fabricante: ('', 801.2561798095703)
```

Figura 6. Prueba de código creado N° 4.

4.3. Pruebas con parámetro –help

Para obtener ayuda sobre la utilización de la herramienta en línea de comandos podemos usar el siguiente comando:



```
PS C:\Andor> python OUILookup.py --help
Use: ./OUILookup --mac <mac> | --arp | [--help]
--mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
--arp: muestra los fabricantes de los host
disponibles en la tabla arp.
--help: muestra este mensaje y termina.
```

Figura 7. Prueba de código creado N° 5.

5. Discusión sobre MACs aleatorias

Las direcciones MAC aleatorias son útiles para mejorar la privacidad/seguridad del usuario, esto debido al constante cambio del identificador único asignado a una interfaz de red en nuestros dispositivos, si bien este concepto por si solo no otorga un nivel de seguridad lo suficientemente alto para proporcionar un servicio con mejor privacidad por si solo, este es uno de los tantos elementos incorporados en el modelo OSI con este objetivo, formado entre ellos un entorno cohesivo que promueva una mayor seguridad para los usuarios.

Tradicionalmente, las direcciones MAC son de carácter estáticas y son asignadas por el fabricante de hardware, es por esto por lo que usar direcciones MAC aleatorias surge como una solución que provee múltiples ventajas al momento de su adopción, algunas de estas pueden ser:

- Seguridad en Redes Públicas: En redes Wi-Fi públicas, la aleatorización de direcciones MAC puede evitar ataques como la falsificación de dirección MAC (MAC spoofing) y la recolección de datos de usuarios.
- Privacidad del Usuario: Las direcciones MAC estáticas permiten rastrear el dispositivo a lo largo del tiempo y a través de diferentes redes. Al utilizar direcciones MAC aleatorias, los dispositivos pueden dificultar el rastreo por parte de terceros.

Pasando al funcionamiento del proceso de aleatorización podemos encontrar la generación de la dirección aleatoria, esta tendrá una temporalidad asignada para poder ser cambiada después de cierto periodo para finalmente conectarse a una red, teniendo la opción de usar una MAC real o aleatoria.

5.1. Referencias bibliográficas

1. RFC 7844 - Anonymity Profiles for DHCP Clients
 - Este documento RFC detalla las técnicas para mantener la privacidad del usuario en redes DHCP, incluyendo el uso de direcciones MAC aleatorias para anonimizar las solicitudes de red.
2. IEEE 802.11-2016 - IEEE Standard for Information Technology
 - Esta referencia proporciona el estándar IEEE para redes Wi-Fi, que incluye las especificaciones sobre el uso de direcciones MAC aleatorias para mejorar la privacidad y seguridad.

5.2. Repositorio GitHub

Se adjunta el enlace al repositorio GitHub asociado al proyecto:
<https://github.com/LucasRojasReyes/Tarea02-Rojas-Reyes-Lucas>.