

Informe Tarea02: Direcciones Ip, Mac y Tabla ARP

Yerko Yuivar Concha, Yerko.yuivar@alumnos.uv.cl

Alex Rodriguez Celis, alex.rodriguez@alumnos.uv.cl

Bastian Puebla Gallardo, bastian.puebla@alumnos.uv.cl

1. Introducción

El script OUILookup.py es una herramienta diseñada para facilitar la identificación y consulta de información relacionada con direcciones IP y direcciones MAC en redes locales. Utilizando Python, el script permite a los usuarios determinar si una dirección IP pertenece a una red específica y obtener datos sobre el fabricante asociado a una dirección MAC a través de una API externa.

2. Descripción del problema y diseño de la solución

Los administradores de red y los usuarios necesitan obtener información sobre direcciones IP y MAC para gestionar mejor sus redes, solucionar problemas de conectividad o simplemente para conocer más sobre los dispositivos que están conectados.

- ⊗ **Consulta de Direcciones IP:** Permite verificar si una dirección IP específica está dentro de una red local definida. Esto ayuda a determinar si un dispositivo está conectado a la misma red.
- ⊗ **Consulta de Direcciones MAC:** Utiliza APIs en línea para obtener información sobre el fabricante de un dispositivo a partir de su dirección MAC. Esto puede ser útil para identificar dispositivos desconocidos en la red.
- ⊗ **Obtención de Tabla ARP:** Extrae información de las conexiones de red actuales, mostrando las direcciones IP y MAC de los dispositivos con los que el sistema tiene comunicación activa.

3. Implementación

La implementación del script se lleva a cabo en Python y se compone de varias funciones clave que manejan la consulta de direcciones IP y MAC, así como la obtención de la tabla ARP. A continuación, se describen las partes más importantes del código.

Definición de la Red Local:

- `red = ipaddress.IPv4Network("192.168.1.0/24")`
- Se define la red local a la que se dirigirán las consultas. En este caso, la red es 192.168.1.0/24, que abarca las direcciones IP desde 192.168.1.1 hasta 192.168.1.254.

Función `obtener_datos_por_ip`:

- `def obtener_datos_por_ip(ip):`
- Esta función verifica si una dirección IP está dentro de la red local. Utiliza la biblioteca `ipaddress` para manejar objetos de dirección IP y determinar su pertenencia a la red.

Función obtener_datos_por_mac:

- def obtener_datos_por_mac(mac)
- Esta función realiza una consulta a una API para obtener el fabricante de un dispositivo a partir de su dirección MAC. Maneja la respuesta de la API y devuelve el nombre del fabricante o un mensaje de error si la consulta falla.

Función obtener_tabla_arp:

- def obtener_tabla_arp():
- Esta función utiliza psutil para obtener la tabla ARP del sistema. Muestra las direcciones IP y MAC de las conexiones activas, lo que permite a los usuarios ver los dispositivos en comunicación.

4. Pruebas

A continuación se mostraran las evidencias de como utilizamos y como respondían los comandos utilizados.

Comandos mostrados en pantalla junto con su respectivo resultado

```
Dirección MAC: 98063C
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de ejecución: 0.6054415702819824 segundos

C:\Users\Lizen\Downloads\Tarea_Red>python tarea2-OUILookup.py --mac 00:01:97:bb:bb:bb

Dirección MAC: 000197
Fabricante: Cisco Systems, Inc
Tiempo de ejecución: 0.6162440776824951 segundos

C:\Users\Lizen\Downloads\Tarea_Red>python tarea2-OUILookup.py --mac b4:b5:fe:92:ff:c5

Dirección MAC:
Fabricante:
Tiempo de ejecución: 0.6263647079467773 segundos

C:\Users\Lizen\Downloads\Tarea_Red>python tarea2-OUILookup.py --mac b4:b5:fe:92:ff:c5

Dirección MAC:
Fabricante:
Tiempo de ejecución: 0.6222126483917236 segundos

C:\Users\Lizen\Downloads\Tarea_Red>python tarea2-OUILookup.py --mac AC:F7:F3:aa:aa:aa

Dirección MAC: ACF7F3
Fabricante: Xiaomi Communications Co Ltd
Tiempo de ejecución: 0.45112156867980957 segundos

C:\Users\Lizen\Downloads\Tarea_Red>
```

Figura 1. Evidencia de Yerko utilizando el cmd.

Resultado de la tabla ARP de Yerko.

Tabla ARP:

Interfaz: 192.168.1.18 --- 0x4			
Dirección de Internet	Dirección física	Tipo	
192.168.1.1	58-ae-f1-52-ca-30	dinámico	
192.168.1.4	b4-ec-ff-03-86-f8	dinámico	
192.168.1.7	b4-ec-ff-03-86-d1	dinámico	
192.168.1.8	b4-ec-ff-01-5a-42	dinámico	
192.168.1.11	86-af-84-8f-14-5c	dinámico	
192.168.1.14	d8-1c-79-ae-5f-bf	dinámico	
192.168.1.15	26-84-4a-70-e1-02	dinámico	
192.168.1.16	64-e0-03-62-c9-9d	dinámico	
192.168.1.17	50-1a-c5-63-dd-02	dinámico	
192.168.1.255	ff-ff-ff-ff-ff-ff	estático	
224.0.0.22	01-00-5e-00-00-16	estático	
224.0.0.251	01-00-5e-00-00-fb	estático	
224.0.0.252	01-00-5e-00-00-fc	estático	
224.0.2.60	01-00-5e-00-02-3c	estático	
239.255.255.250	01-00-5e-7f-ff-fa	estático	
255.255.255.255	ff-ff-ff-ff-ff-ff	estático	

Interfaz: 172.18.64.1 --- 0x11			
Dirección de Internet	Dirección física	Tipo	
172.18.79.255	ff-ff-ff-ff-ff-ff	estático	
224.0.0.22	01-00-5e-00-00-16	estático	
224.0.0.251	01-00-5e-00-00-fb	estático	
239.255.255.250	01-00-5e-7f-ff-fa	estático	

Interfaz:		192.168.1.18	
192.168.1.1	58-ae-f1-52-ca-30		
192.168.1.4	b4-ec-ff-03-86-f8		
192.168.1.7	b4-ec-ff-03-86-d1		
192.168.1.8	b4-ec-ff-01-5a-42		
192.168.1.11	86-af-84-8f-14-5c		
192.168.1.14	d8-1c-79-ae-5f-bf		
192.168.1.15	26-84-4a-70-e1-02		
192.168.1.16	64-e0-03-62-c9-9d		
192.168.1.17	50-1a-c5-63-dd-02		
192.168.1.255	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		
224.0.2.60	01-00-5e-00-02-3c		
239.255.255.250	01-00-5e-7f-ff-fa		
255.255.255.255	ff-ff-ff-ff-ff-ff		
Interfaz:		172.18.64.1	
172.18.79.255	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		
239.255.255.250	01-00-5e-7f-ff-fa		

Figura 2. Evidencia de la tabla ARP de Yerko.

Evidencia de Alex Rodriguez:

```
PS C:\Users\plugs> & C:/Users/plugs/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/plugs/Desktop/OUILookup.py
usage: OUILookup.py [-h] [--ip IP] [--mac MAC] [--arp]
```

Consulta el fabricante de una tarjeta de red dada su dirección MAC o IP.

options:

```
-h, --help  show this help message and exit
--ip IP     IP del host a consultar.
--mac MAC   MAC a consultar.
--arp       Muestra los fabricantes de los host disponibles en la tabla ARP.
```

```
PS C:\Users\plugs> cd C:\Users\plugs\Desktop
```

```
PS C:\Users\plugs\Desktop> python OUILookup.py --mac 98:06:3c:92:ff:c5
```

Dirección MAC: 98063C

Fabricante: Samsung Electronics Co.,Ltd

Tiempo de ejecución: 0.9839680194854736 segundos

```
PS C:\Users\plugs\Desktop> python OUILookup.py --mac 00:01:97:bb:bb:bb
```

Dirección MAC: 000197

Fabricante: Cisco Systems, Inc

Tiempo de ejecución: 1.3378779888153076 segundos

```
PS C:\Users\plugs\Desktop> python OUILookup.py --mac b4:b5:fe:92:ff:c5
```

Dirección MAC:

Fabricante:

Tiempo de ejecución: 1.328620195388794 segundos

```
PS C:\Users\plugs\Desktop> python OUILookup.py --mac 00:E0:64:aa:aa:aa
```

Dirección MAC: 00E064

Fabricante: SAMSUNG ELECTRONICS

Tiempo de ejecución: 0.708634614944458 segundos

```
PS C:\Users\plugs\Desktop> python OUILookup.py --mac AC:F7:F3:aa:aa:aa
```

Dirección MAC: ACF7F3

Fabricante: Xiaomi Communications Co Ltd

Tiempo de ejecución: 0.8213498592376709 segundos

Figura 3. Evidencia de alex utilizando la terminal de VSC.

Tabla ARP de alex rodriguez:

```
PS C:\Users\plugs\Desktop> python OUILookup.py --arp
```

Tabla ARP:

Interfaz: 192.168.0.51 --- 0xa

Dirección de Internet	Dirección física	Tipo
192.168.0.1	78-6a-1f-5b-40-6b	dinámico
192.168.0.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
224.0.2.60	01-00-5e-00-02-3c	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

Interfaz: 192.168.0.51

192.168.0.1 78-6a-1f-5b-40-6b

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

224.0.2.60 01-00-5e-00-02-3c

239.255.255.250 01-00-5e-7f-ff-fa

255.255.255.255 ff-ff-ff-ff-ff-ff

```
PS C:\Users\plugs\Desktop> █
```

Evidencia Bastián Puebla:

```
Dirección MAC: 000197
Fabricante: Cisco Systems, Inc
Tiempo de ejecución: 0.418734073638916 segundos
[bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 % python3 message.py --mac b4:b5:fe:92:ff:c5 ]

Dirección MAC:
Fabricante:
Tiempo de ejecución: 0.8363127708435059 segundos
[bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 % python3 message.py --mac 00:E0:64:aa:aa:aa ]

Dirección MAC: 00E064
Fabricante: SAMSUNG ELECTRONICS
Tiempo de ejecución: 0.8233652114868164 segundos
[bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 % python3 message.py --mac AC:F7:F3:aa:aa:aa ]

Dirección MAC: ACF7F3
Fabricante: Xiaomi Communications Co Ltd
Tiempo de ejecución: 0.8201909065246582 segundos
[bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 % python3 message.py --mac 48-E7-DA ]

Dirección MAC: 48E7DA
Fabricante: AzureWave Technology Inc.
Tiempo de ejecución: 0.629105806350708 segundos
bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 %
```

Figura 5. Evidencia Bastian pueblas a través de la terminal de Linux.

Tabla ARP de Bastian puebla.

```
[bastianignaciopueblagallardo@MacBook-Air-de-sigrid TAREA2 % python3 message.py --arp

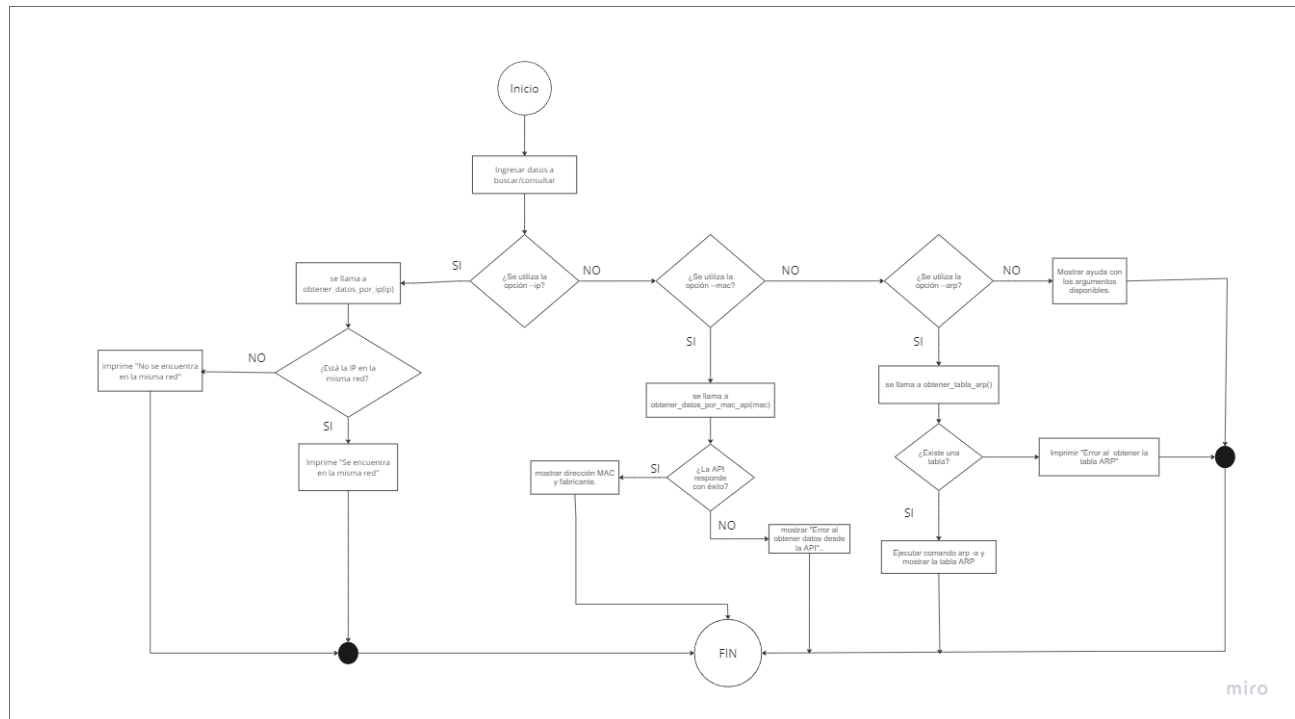
Tabla ARP:
? (192.168.3.1) at 24:81:c7:a7:d1:83 on en0 ifscope [ethernet]
? (192.168.3.4) at 7e:1c:c3:a9:71:d6 on en0 ifscope [ethernet]
? (192.168.3.10) at 24:a1:60:37:b:b4 on en0 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
```

Figura 6. Evidencia de Bastián puebla junto al comando de la tabla ARP.

4.2. Respuesta a la pregunta (pdf 4.4).

Las direcciones MAC aleatorias son identificadores temporales utilizados por dispositivos para mejorar la privacidad del usuario al conectarse a redes Wi-Fi. Al cambiar periódicamente la dirección MAC, se dificulta el rastreo y la identificación del dispositivo por parte de terceros, lo que ayuda a proteger la información personal y los hábitos de conexión. Esta funcionalidad es especialmente útil en redes públicas, aunque puede presentar desafíos en redes que utilizan filtrado por dirección MAC.

4.3 Diagrama de flujo del código.



5. Discusión y conclusiones

La creación del script para consultar direcciones IP y MAC ha sido un proceso enriquecedor que ha permitido explorar diversas funcionalidades de Python

Las consultas a direcciones IP y MAC proporcionan datos valiosos que pueden ser utilizados tanto para la gestión de redes como para la resolución de problemas. La inclusión de la tabla ARP es particularmente útil para visualizar conexiones activas, lo que facilita la identificación de dispositivos desconocidos.

Durante el desarrollo, se presentaron dificultades, como el manejo de errores en consultas a APIs (ya que nos apareció el error 404 page not found) y la necesidad de validar formatos de entrada. Estos problemas se resolvieron mediante la implementación de controles adecuados y manejo de excepciones.

6. Referencias

- [1] Direcciones Mac aleatorias en dispositivos Apple
- [2] Uso de direcciones MAC aleatorias - BSAM-DI-06
- [3] Spectrum.net