

Tarea II : OUILookup

Felipe Caru Prado, felipe.caru@alumnos.uv.cl .

1. Introducción

La tarea consiste en desarrollar una herramienta de línea de comandos en Python llamada "OUILookup" se escogió el sistema operativo Windows, el programa permite consultar el fabricante de una tarjeta de red a partir de su dirección MAC o dirección IP. Para lograrlo, se debe utilizar una base de datos parametrizada como variable global. Esta herramienta será de gran utilidad para identificar el fabricante de dispositivos de red en una red local. Además, se incluirán opciones para mostrar información de la tabla ARP y proporcionar ayuda con la opción "--help".

2. Bibliotecas utilizadas y archivos

El código utiliza las siguientes bibliotecas:

- **subprocess**: Esta biblioteca permite ejecutar comandos del sistema operativo. Se utiliza para obtener la tabla ARP.
- **getopt**: Esta biblioteca permite analizar las opciones de la línea de comandos. Se utiliza para obtener los argumentos de entrada del programa.
- **sys**: Esta biblioteca proporciona acceso a variables y funciones del sistema operativo. Se utiliza para obtener el número de argumentos de entrada del programa y para terminar el programa.

El archivo utilizado para las pruebas es el definido en el informe de trabajo dado por el profesor:

<https://github.com/boundary/wireshark/blob/master/manuf>

3. Funcionamiento

El esqueleto proporcionado por la descripción del trabajo, me permitió simplificar la organización del código, por lo tanto mi programa presenta las siguientes 6 funciones:

3.1 cargar_archivo(): Esta función se encarga de cargar datos desde un archivo llamado "DBOUI.txt". Estos datos representan una base de datos de direcciones MAC y sus descripciones.

3.2 obtener_datos_por_ip (ip): Toma una dirección IP como entrada y busca su correspondiente dirección MAC en la tabla ARP. Luego, busca la dirección MAC en la base de datos y muestra el fabricante si se encuentra.

3.3 obtener_datos_por_mac(direccion_mac): Toma una dirección MAC como entrada y busca el fabricante en la base de datos. Si no se encuentra, nuestra "Not found".

3.4 obtener_tabla_arp(): Utiliza el comando 'arp -a' para obtener la tabla ARP del sistema y luego procesa la salida para obtener un diccionario que relaciona las direcciones IP con sus correspondientes direcciones MAC.

3.5 mostrar_ayuda(): Muestra un mensaje de ayuda que describe cómo usar el script y sus opciones.

3.6 main(argv): El punto de entrada del programa. Procesa los argumentos de línea de comandos y realiza acciones basadas en las siguientes opciones proporcionadas por el usuario.

- "--ip" busca información basada en una dirección IP.
- "--mac" busca información basada en una dirección MAC.
- "--arp" muestra información sobre las direcciones MAC en la tabla ARP.
- "--help" muestra un mensaje de ayuda.

4. Resultados

Dentro de mi entorno local, realice pruebas con los siguientes resultados:

4.1. - - arp : al ejecutar el programa con el comando `--arp` muestra correctamente las ip y mac asociadas al entorno local donde se ejecuta el programa, cabe recalcar que aparecen fabricantes no encontrados, ya que pueden ser macs inexistentes o no registradas en la base de datos

```
PS C:\Users\felip> python OUILookup.py --arp
<IP>-----<Dirección MAC>-----<FABRICANTE>
Dirección      DE, No encontrado
192.168.0.1     48:D3:43:B5:46:F0, No encontrado
192.168.0.252   00:00:CA:01:02:03, ArrisInt          # ARRIS International
224.0.0.22      01:00:5E:00:00:16, No encontrado
224.0.0.251     01:00:5E:00:00:FB, No encontrado
224.0.0.252     01:00:5E:00:00:FC, No encontrado
255.255.255.255 FF:FF:FF:FF:FF:FF, Broadcast
PS C:\Users\felip> python OUILookup.py --mac 00:00:CA:01:02:03
```

4.2. - -ip : al ejecutar el programa con el comando `--ip` muestra correctamente la ip y su dirección mac .

- Caso 1 : Caso ip que pertenezca a su misma red

```
PS C:\Users\felip> python OUILookup.py --ip 192.168.0.252
Dirección MAC: 00:00:CA:01:02:03 ---- Fabricante: ArrisInt
```

- Caso 2 : Caso ip que NO pertenezca a su misma red

```
PS C:\Users\felip> python OUILookup.py --ip 192.168.30.0
Error: IP está fuera de la red del host
No comienza con 192.168.0.
```

4.3. - -mac : Al ejecutar el programa con el comando `--mac` muestra correctamente la mac y el fabricante, se utiliza la misma mac que en el ejemplo del comando `--ip` ya que es el único que tiene su empresa de fabricación .

```
PS C:\Users\felip> python OUILookup.py --mac 00:00:CA:01:02:03
Dirección MAC: 00:00:CA ---- Fabricante: ArrisInt # ARRIS International
PS C:\Users\felip>
```

4.4. - -help : Al ejecutar el programa con el comando `--help` muestra correctamente el mensaje de ayuda para que el usuario pueda comprender el funcionamiento del programa.

```
PS C:\Users\felip> python OUILookup.py --help
Uso: python OUILookup.py --ip <IP> | --mac <IP> | --arp | [--help]
--ip : IP del host a consultar.
--mac: Dirección MAC a consultar. P.e. aa:bb:cc:00:00:00.
--arp: muestra los fabricantes de los hosts disponibles en la tabla ARP.
--help: muestra este mensaje y termina.
PS C:\Users\felip>
```

5. Conclusiones

La herramienta "OUILookup" desarrollada en Python elegida para Windows resuelve la problemática de identificar el fabricante de dispositivos de red a partir de direcciones MAC e IP. Proporciona una solución eficiente y útil para los administradores de red y usuarios que necesitan rastrear y gestionar los dispositivos en una red local. Con la capacidad de consultar tanto direcciones MAC como direcciones IP y mostrar información de la tabla ARP, esta herramienta mejora la visibilidad y la gestión de la red. La implementación de la base de datos de fabricantes como variable global asegura que la herramienta esté actualizada y sea capaz de identificar una amplia gama de dispositivos.