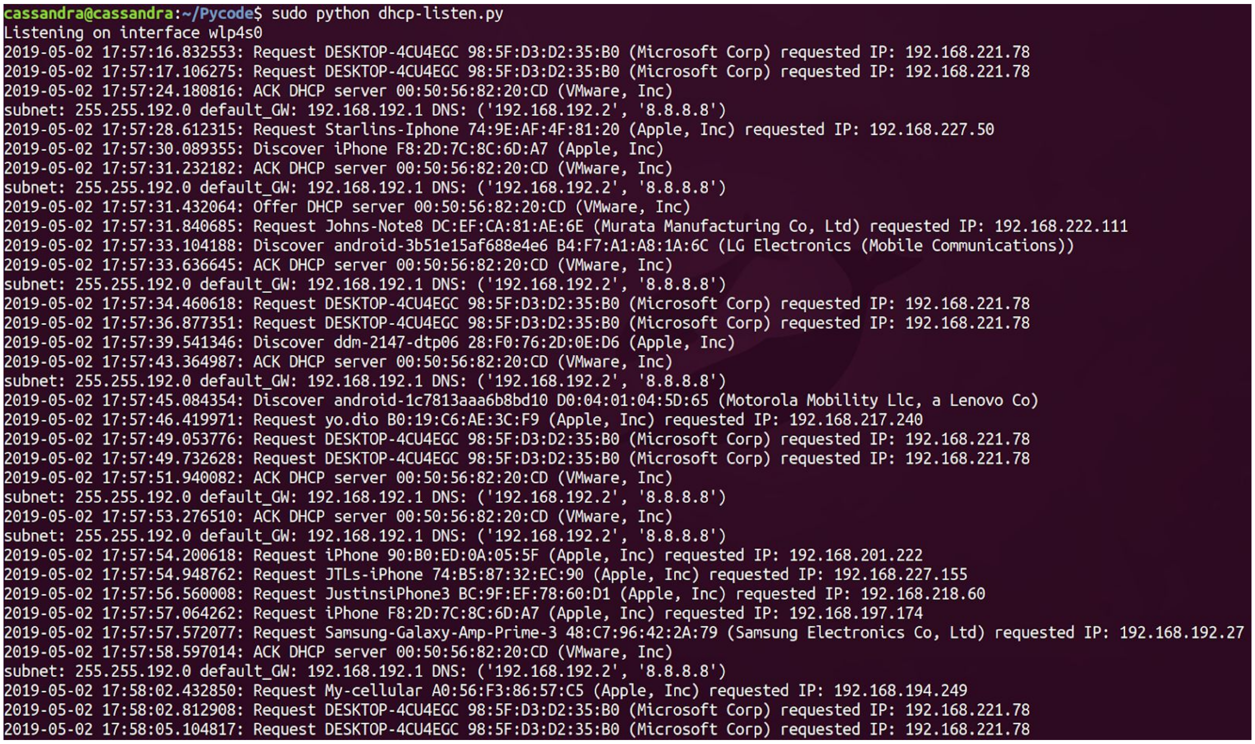
# About

The Dynamic Host Configuration Protocol (DHCP) is insecure and introduces an additional attack vector to the LAN. An analysis of DHCP traffic allows an eavesdropper to obtain the configuration information of devices within a broadcast domain. The Python programming language has several libraries that make packet capture and analysis possible.

This security tool was developed to monitor DHCP traffic and detect DHCP attacks. The tool is written in Python and utilizes the scapy library to filter, collect, and examine DHCP packets. The tool reports data on the timestamp, DHCP message type, device hostname, MAC address, and vendor, and, if applicable, the requested IP address. This information is displayed to the terminal live and written to a log file. From the log file, it is possible to map the exchange between servers and clients and see how it changes based on various conditions. The tool can detect anomalous and malicious DHCP activity.

The tool exploits the insecure nature of DHCP traffic and collects as much interesting information as possible. The Python language in combination with the scapy library allows for packet capture and inspection. The tool passively listens for DHCP traffic. Once a packet is found, the tool searches through its contents and extracts the DHCP message type, the source device's hostname, MAC address, and requested IP if applicable. With knowledge of the source device's MAC address, the device's vendor can be determined. All acquired device information is sent to the terminal and written to a log file for later use.

Example Output:



```
cassandra@cassandra:~/Pycode$ sudo python dhcp-listen.py
Listening on interface wlp4s0
2019-05-02 17:57:16.832553: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:17.106275: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:24.180816: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:28.612315: Request Starlins-Iphone 74:9E:AF:4F:81:20 (Apple, Inc) requested IP: 192.168.227.50
2019-05-02 17:57:30.089355: Discover iPhone F8:2D:7C:8C:6D:A7 (Apple, Inc)
2019-05-02 17:57:31.232182: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:31.432064: Offer DHCP server 00:50:56:82:20:CD (VMware, Inc)
2019-05-02 17:57:31.840685: Request Johns-Note8 DC:EF:CA:81:AE:6E (Murata Manufacturing Co, Ltd) requested IP: 192.168.222.111
2019-05-02 17:57:33.104188: Discover android-3b51e15af688e4e6 B4:F7:A1:A8:1A:6C (LG Electronics (Mobile Communications))
2019-05-02 17:57:33.636645: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:34.460618: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:36.877351: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:39.541346: Discover ddm-2147-dtp06 28:F0:76:2D:0E:D6 (Apple, Inc)
2019-05-02 17:57:43.364987: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:45.084354: Discover android-1c7813aaa6b8bd10 D0:04:01:04:5D:65 (Motorola Mobility Llc, a Lenovo Co)
2019-05-02 17:57:46.419971: Request yo.dio B0:19:C6:AE:3C:F9 (Apple, Inc) requested IP: 192.168.217.240
2019-05-02 17:57:49.053776: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:49.732628: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:57:51.940082: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:53.276510: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:57:54.200618: Request iPhone 90:B0:ED:0A:05:5F (Apple, Inc) requested IP: 192.168.201.222
2019-05-02 17:57:54.948762: Request JTLs-iPhone 74:B5:87:32:EC:90 (Apple, Inc) requested IP: 192.168.227.155
2019-05-02 17:57:56.560008: Request JustinsiPhone3 BC:9F:EF:78:60:D1 (Apple, Inc) requested IP: 192.168.218.60
2019-05-02 17:57:57.064262: Request iPhone F8:2D:7C:8C:6D:A7 (Apple, Inc) requested IP: 192.168.197.174
2019-05-02 17:57:57.572077: Request Samsung-Galaxy-Amp-Prime-3 48:C7:96:42:2A:79 (Samsung Electronics Co, Ltd) requested IP: 192.168.192.27
2019-05-02 17:57:58.597014: ACK DHCP server 00:50:56:82:20:CD (VMware, Inc)
subnet: 255.255.192.0 default_GW: 192.168.192.1 DNS: ('192.168.192.2', '8.8.8.8')
2019-05-02 17:58:02.432850: Request My-cellular A0:56:F3:86:57:C5 (Apple, Inc) requested IP: 192.168.194.249
2019-05-02 17:58:02.812908: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
2019-05-02 17:58:05.104817: Request DESKTOP-4CU4EGC 98:5F:D3:D2:35:B0 (Microsoft Corp) requested IP: 192.168.221.78
```

# Usage

Because of its convenience, DHCPv4 is a widely implemented protocol that operates on the majority of IPV4 networks. DHCP offers an automated way for clients to obtain an IP address, subnet mask, default gateway, and DNS information without administrator intervention. This tool can be run on any network configured with DHCPv4 to report information on DHCP servers and clients operating on that broadcast domain.

The insecure nature of DHCP leaves it vulnerable to various attacks: eavesdropping, starvation, masquerading, and man-in-the-middle. The tool collects and reports information on devices operating within a broadcast domain. With this information, the script can detect and report on various DHCP attacks in progress. A sudden high volume of DISCOVER and REQUEST messages tip off a starvation attack. Similarly, examination of the subnet mask, default gateway, and DNS information being provide to clients would alert admins of a rouge server, or man-in-the-middle in progress.

Alternatively, this information enables an adversary to build profiles on clients containing the devices hostname, MAC address, vendor, and potentially their IP address. Using these profiles, an adversary can target specific users, and determine device-specific vulnerabilities based on the manufacturer. They could also target the DHCP server by performing a starvation attack. With the data provided by this tool, an adversary could also launch a masquerading attack where it pretends to be a DHCP server and hands out illegitimate configuration parameters to clients. A successful masquerading attack enables adversaries to man-in-the-middle clients. By passing their IP addresses in place of the legitimate gateway IP address they can redirect traffic destined to an external network to themselves.

This tool can be run on any network configured with DHCPv4 to report information on DHCP servers and clients operating on that broadcast domain.

# Installation/Step up

Required libraries include sys, time, pandas, argparse, and scapy.
If you are missing any of these libraries, simply use pip to install them (ex. pip install scapy)

Required files include macaddress.io-db.csv originally pulled from https://macvendors.com. Place this file in the same directory as this script.

To use the script follow the usage guide described with the -h or --help option.
Ex: python dhcp-listen.py -f DHCP_capture.txt -i wlp2s0

# Special note

The current logical execution of the tool leads to the potential loss of DHCP packets. After a DHCP packet is captured, the tool executes a process that extracts valuable information from the packet. During the analysis process, the tool briefly stops listening for DHCP traffic. The analysis process takes an average of 4.012 nanoseconds. If any DHCP traffic were to be transmitted during this time, it would go unseen.

To remedy this, an new implementation would use the multiprocessing library to enable the capture and analyze functionality to be executed simultaneously. Concurrent execution allows for continuous packet capture, and live analysis and output to the terminal.

# Tips

How to find your interface names.
Linux: use either ip -a or ifconfig in the terminal
Windows: use ipconfig in command prompt