**LAB-04: ARP Cache Poisoning using Scapy**

**1. Lab Overview**

ARP cache poisoning is one of the most popular ways of doing a MITM attack on a local area network. This article is going to demonstrate how to build a python program to poison the ARP cache of the target and the gateway in order to perform a successful MITM attack.

In order to build the program, we need to go through the steps of ARP Cache Poisoning, and then I will demonstrate it, using Scapy. To demonstrate ARP Poisoning, I have the attacker’s machine on the VirtualBox running Kali Linux in Bridged Adapter with the Target machine running Windows. Both the machines are connected on a WLAN.

As the name of the attack suggests, we will be poisoning the ARP cache of the target. Each machine on the LAN has a local ARP table(cache) that it maintains. This table is formed by the various ARP responses that the machine receives for different ARP requests.

For example, Machine A (10.0.2.5) wants to communicate to Machine B (10.0.2.6). In order to communicate, Machine A requires the MAC address of Machine B. So, Machine A searches his ARP table(cache) if he could find the MAC address associated with the IP address 10.0.2.6. If he does, well and good he can send the packet to Machine B, else Machine A will send an ARP broadcast message. The ARP broadcast is directed to ff:ff:ff:ff:ff:ff. The request message will travel across the network to every machine asking if that machine’s IP address matches 10.0.2.6. When machine B finds the ARP request, it sends an ARP response to Machine A (10.0.2.5) telling what its MAC address is. Machine A writes this to its local ARP table.

**2. Lab Environment**

This LAB setup consists of the target machine (192.168.43.65), the attacker machine (192.168.43.220) and the Gateway(192.168.43.1). The attacker machine will be used to do the poisoning on the target and the gateway, to successfully carry out the MITM sniffing attack.

**Note:** Ensure packet forwarding is enabled. To enable it use the following command: **echo 1 > /proc/sys/net/ipv4/ip\_forward.**

Windows 10 Enterprise VirtualBox/VM free image:

[https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdeveloper.microsoft.com%2Fen-us%2Fwindows%2Fdownloads%2Fvirtual-machines%2F&data=04%7C01%7Camailewa%40stcloudstate.edu%7Cf5f57b3782464ea3aafe08d8ce3decc0%7C5011c7c60ab446ab9ef4fae74a921a7f%7C0%7C0%7C637486113010910893%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=LspAHb5LQsrg1pHJ1jb%2BQtm1NTWvFxKyfROvAjfjkZs%3D&reserved=0)

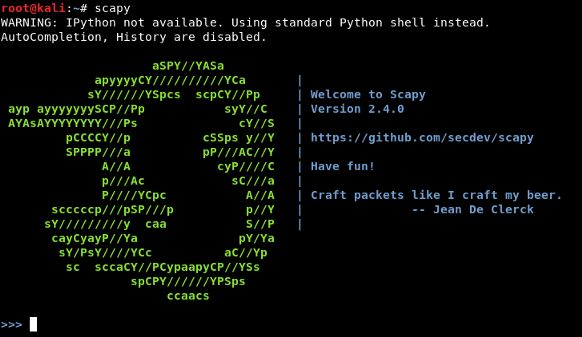
Scapy Download and Installation: <https://scapy.readthedocs.io/en/latest/installation.html>

**3. Lab Task - 1**

**Part I: Introduction to Scapy**

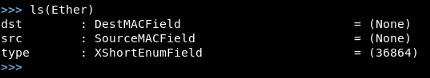
Scapy is a Python program that enables the user to send, sniff and dissect and forge network packets. This capability allows construction of tools that can probe, scan or attack networks. To know how to install it, one can go to aforementioned Scapy Download and Installation link.

**Step 1:** Run scapy in Kali Linux by typing **scapy** in a terminal.

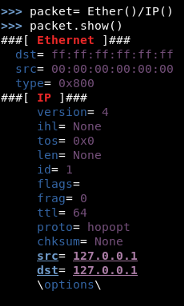
.

Scapy can be used to sniff, dissect and forge packets. This LAB will be about ARP poisoning, and will be dealing only with forging of packets.

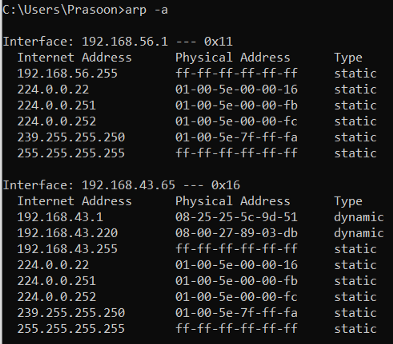
**Step 2:** To see what are the fields of a layer, use the **ls()** function.



**Step 3:** To create a packet, append two layers together using the **“/” operator**.



**Step 4:** To see the ARP table in both Windows and Linux, use the command **arp -a**.



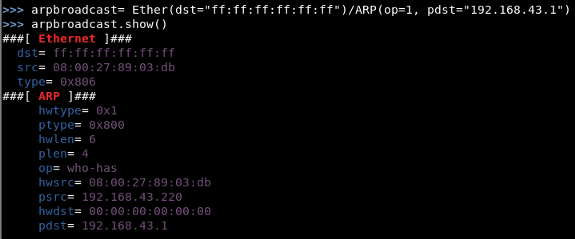
**Part II: Finding out the MAC address of the target and the Gateway**

Send a broadcast message for the target and the gateway inorder to find their MAC address. Design the ARP broadcast request for IP address= 192.168.43.1(gateway).

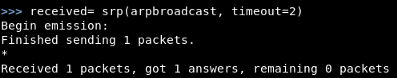
**Note**: In ARP layer, **hwsrc** and **hwdst** represent **MAC address of source and destination**respectively, while **psrc**and **pdst** represent the**IP address of source and destination respectively.**

**Step 1:** On a Kali Linux terminal run scapy.

**Step 2:** ARP broadcast packet in Scapy by using the following commands:



**Step 3:** Use **srp()** to send the packet and then receive the response packet:



**Step 4:** Here is the MAC address:

Image for post

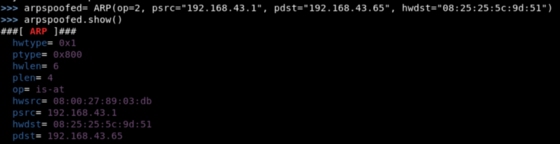
**Step 5:** Similarly, do the same thing to find the MAC address for the target(192.168.43.65).

Image for post

**Part III: Sending false ARP response packets to both the target and the gateway.**

The false ARP response to the target will contain the pdst= ‘192.168.43.65’ hwdst= ‘08:25:25:5c:9d:51' and psrc= ‘192.168.43.1’. By default, this packet would have the attacker’s MAC address. Thus when the target gets the packet it updates its ARP table with rogue MAC address associated with the gateway’s IP address. To ensure that the poisoning is not cured, continuously send the ARP responses, which will be done in the python script.

**Step 1:** Spoofed packet for 192.168.43.65:



**Step 2:** Send the spoofed packet, the response packet is not needed so use send():

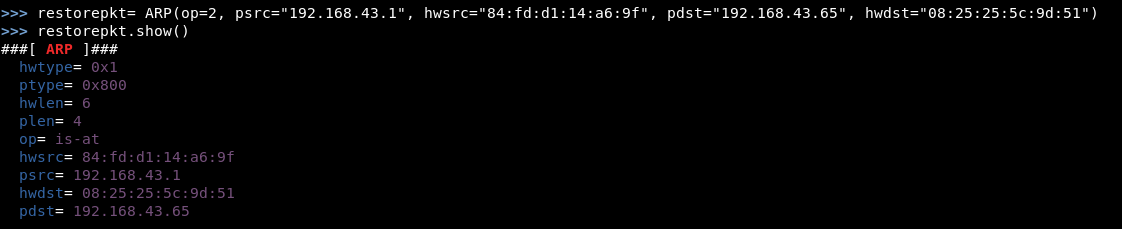


**Step 3:** Similarly, craft a packet for the gateway (192.165.43.1, 84:fd:d1:14:a6:9f ) by spoofing the psrc as “192.168.43.65”.

**Part III: Once the attack is done. Remember to restore the ARP tables of the machines.**

To restore the ARP tables, craft the packets that originally should have been used.

**Step 1:** From the gateway to the target:

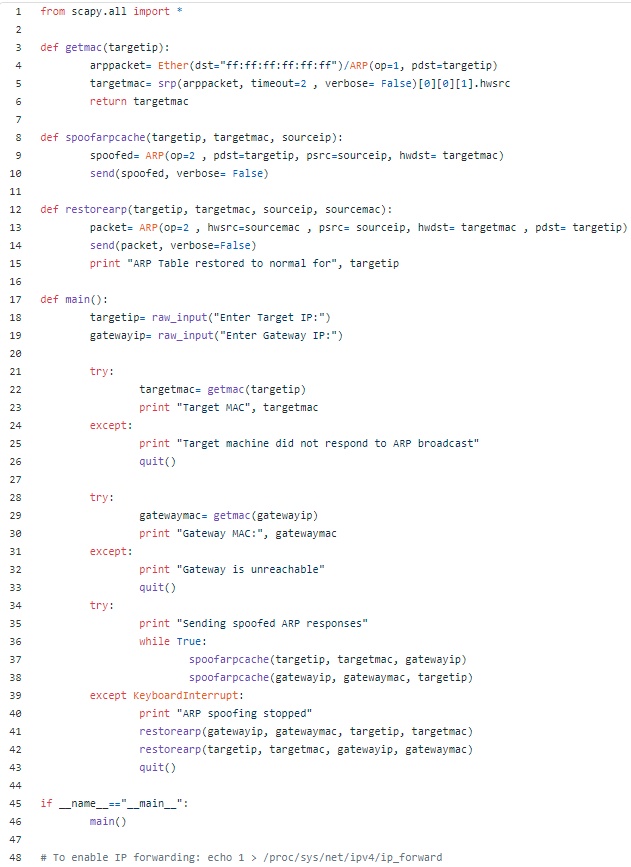
****

**Step 2:** From the target to the gateway:

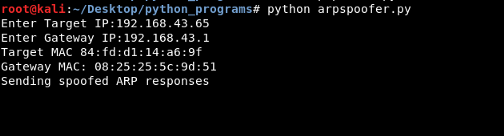


**Part IV: Automate the Whole Process Using Python Script**

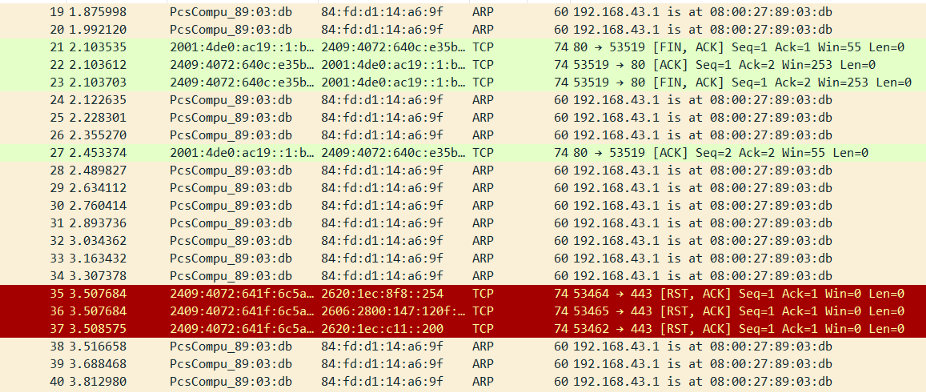
**Step 1:** Create this python script:



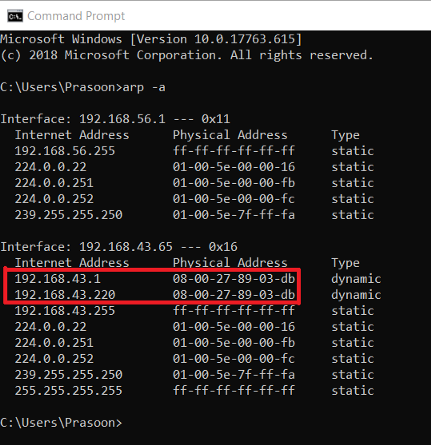
**Step 2:** Run the script:



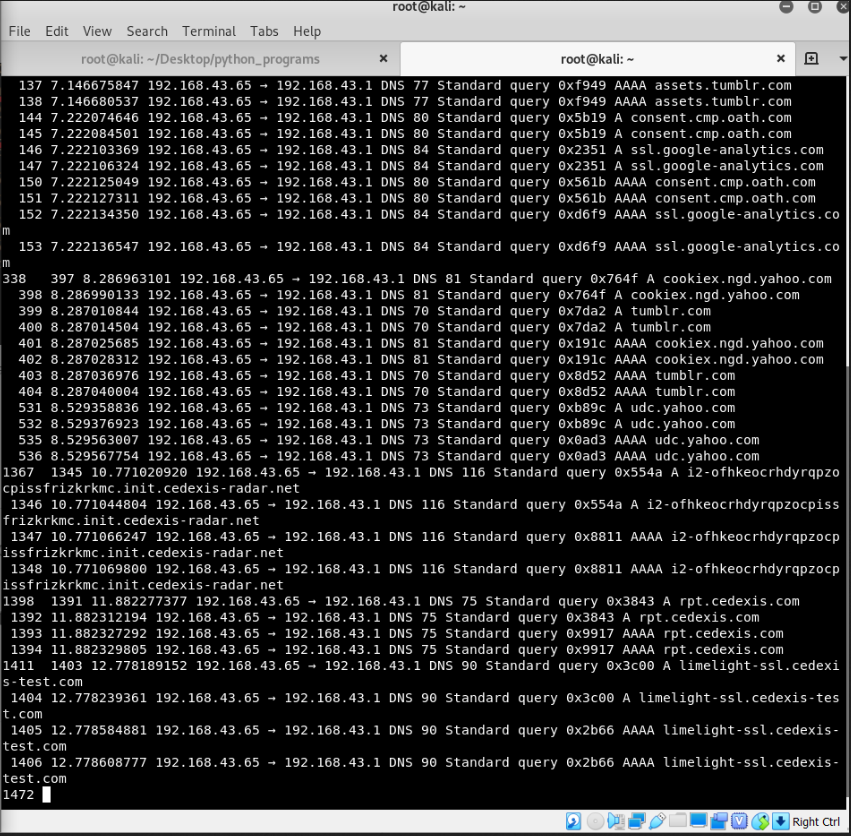
**Step 3:** Traffic on the target machine:



**Step 4:** Check the target ARP cache:



You will also notice that the MITM is also working, if you does **tshark -i <interface> | grep DNS**, on the attacker's machine, one can see the DNS requests of the target going through.



**4. Lab Task – 2: Packet sniffing with Wireshark**

One of the most well-known packet sniffers is called Wireshark® (formerly Ethereal®). It is a powerful tool that can capture, filter, and analyze network traffic. It can promiscuously capture traffic on both wired and wireless networks. It is used by security and networking professionals to troubleshoot networking problems.

In this project, you will install Wireshark, capture packets, and look at the contents of a packet. When the device is placed correctly, a network administrator can use Wireshark to see all the traffic coming into and out of a network. Network administrators can, among other things, see which hostnames are being requested and who is requesting them. Surfing the web is not anonymous.

1. Download Wireshark from http://www.wireshark.org/download.html.

2. Click Download Windows Installer. (Download the latest stable release.)

3. Click Save.

4. Save the file in your download folder.

5. If the program doesn’t automatically open, browse to your download folder.

6. Double-click Wireshark-setup-3.0.3.exe. (The software version numbers will be slightly different as newer versions are released.)

7. Finish the installation process

8. Double-click the Wireshark icon on your desktop. (You can also access it through your Start menu.)

9. Click Interface List. (This will display a list of all available network interfaces on your computer. You will want to want to note the description and IP address of the interface with the most traffic. You will need to select this interface in the following steps.)

10. Note the interface with the most traffic. (You will select this interface in the following steps. If there are duplicate names for the Network Interface Card (NIC), you can use the last three or four values of the MAC address to identify the appropriate NIC.)

11. Close the Capture Interfaces window.

12. Click Capture, and Options.

13. Select your Network Interface Card if it is not already selected.

14. Take a screenshot.

15. Close *all* other programs you currently have open except your word processing program (Microsoft Word, LibreOffice Writer®, etc.).

16. Click Start.

17. Let it run for 10 seconds.

18. While you are waiting open a web browser and go to www.google.com.

19. Return to your Wireshark window.

20. In the file menu click Capture and Stop (or use the keyboard shortcut—Ctrl+E).

21. Scroll up until you see a green and blue area. (These are the packets you captured when you requested Google’s main page.)

22. Take a screenshot.

23. Scroll down until you see a line that has GET / HTTP/1.1. (You may have to try more than one until you get to the packet that shows “www.google.com” in the bottom pane.)

24. Select that row.

25. In the bottom pane, you will see a bunch of numbers to the left. (It’s the packets contents in hexadecimal.) Just to the right you will see the content of the packet in a column.

26. Select the text: www.google.com.

27. Take a screenshot.

**5. Lab Task – 3: THOUGHT QUESTIONS**

1. Why does your computer send so many packets? Why not send just one *big* packet?

2. What do SYN, ACK, FIN, and GET mean?

3. Why do some packets have sequence numbers?

4. Why does your computer send packets to the webserver that you requested data from?

5. What do the different colors in the Wireshark packet capture listing mean?

6. Why would your computer get packets that are addressed to another computer?

7. How many packets does your computer send/receive in a single mouse click when you visit a website?

8. Could you organize or filter the traffic to make it easier to understand?

9. How could blocking all ICMP traffic protect you?

10. Could you still access some websites with your Port 80 rule enabled? Why?

11. Why would you want to allow incoming (not outgoing) Port 443, but block incoming Port 80?

12. Could malware rename itself in order to get through a firewall? Why would this work?

**6. Submission Instructions**

1. **Complete** all the tasks assigned in the project. **Take** screenshots of all major steps involved in this LAB. **Paste** the screenshots into a MS-Word file and if available, **add** the source code of the entire program **(Not all labs have codes)**. **Rename** the MS-Word report file into “LAB-04-Group\_XX” format.
2. You are not always required to follow my steps as it is, you can come up with your own ideas in your own ways to solve the same problem.
3. Try your best to prepare well organized lab report with step by step description for each screenshot.
4. Submit your LAB report to the D2L “LAB-04” drop-box by only one member of each group within one week from the assigned date.

**NOTE-1:** Please add, 1.All group member’s names (LAST\_NAME, FIRST\_NAME) and 2.Page numbers in to the report. If you would, you can create a cover page for each LAB, but not necessary.

**NOTE-2:** Please add all group member’s contributions to complete and submit this lab as a percentage as shown below at the end of the report. (Before submitting to D2L all of the group members must be aware their reported contribution as a percentage in the lab report)

**Example:**

Member-01: 100%

Member-02: 75%

Member-03: 100%

Member-04: 50%