ARP Spoofing Attack and it's Detection

Yogendra R Bijapur

July 2022

1 Introduction

Today, each and everyone are using internet which helps us to do our tasks in a blink of an eye. As internet is growing day by day number of users are also increasing without any surprise hackers are also increasing. Now a days we can see public WIFI every where and people are using them without any fear. But this WIFI's are sniffed very easily just by entering a single command. In this paper I am going discus about same type of attack which is called as ARP spoofing or ARP poisoning attack. I am going to discus how this attack is performed and will provide automated python script which will perform ARP spoofing attack just by providing a target IP's. I will be also providing the ARP spoofing attack and ARP spoofing Detection model.

2 Related Work

Buzura et al. [2] explained how the APR spoofing is performed. They explained the SDN architecture and why it is important in the protection of the network. They have proposed the alternative idea of an ARP spoofing attack in order to reduce the chances of detection. Furthermore, they explained two algorithms on how they can detect the attacker. Then they used the Mininet environment to perform a testing scenario and explained the commands used in the Mininet environment. Finally, they showed the performers by using graphs. Balagopal et al. [1] improved solution for detecting and mitigating ARP poison routing for SDN networks. This module can identify ARP poison routing and overcome it without degrading the network performance. The module not only successfully blocks the malicious host from further attacks but also recovers the ARP Cache table of the victim, so that it can continue functioning within the network without rebooting . After securing the victim device from APR poisoning, the bandwidth of the device is comparatively the same as pre-attack and post-attack. An ARP spoofing attack is one of the biggest threats nowadays, and is very easy to perform just by poisoning the ARP table. Xia et al.[10] proposed a novel mechanism to prevent the ARP spoofing attack oriented towards the OpenFlow platform, as it boosts the SDN of the network. KHALID et al.[4] have discussed the ARP spoofing and have presented the deep study on existing

solution and SDN environments. They proposed a model which will be extended to the SDN controller to detect ARP spoofing. The simulation results showed that the proposed mechanism is robust against ARP spoofing attacks.

Pingle et al.[8] explained the Man-in-the-middle (MITM) Attack, which is one of the biggest threats in this modern era. This paper presents one such successful implementation of a popular network attack, which is the ARP poisoning or ARP spoofing attack, by using the Ettercap tool. When an attacker performs the MITM attack, they can sniff the victim's network but the HTTPS protocol is encrypted so that they can't analyze their packets. In order to convert HTTPS to HTTP, they performed another attack called SSL Strip. The Address Resolution Protocol is one of the most used protocols. The ARP protocol helps us to map IP and MAC addresses. ARP poisoning is an ARP spoofing attack, carried out over a LAN (local area network). Majumdar et al.[5] developed an ARP spoofing attack tool with a Python script. They proposed a detection algorithm to detect the ARP poisoning attack and implemented the algorithm in a Python script with the scapy library. The algorithm is based on optimizing the real MAC address given the ARP packet sniffed MCA address. Finally, they proposed the prevention method and implemented it. Cyber-crimes have become the biggest threat nowadays, and they are very dangerous. The Man in the Middle [MITM] attack is one of them, by which an attacker can sniff the traffic of two people who are communicating through the internet. It can be achieved by poisoning the ARP table by changing the IP and MAC, which is known as ARP spoofing or ARP poisoning. According to the author Morsy et al. [6] there are many detection methods. Some of these approaches depend on the centralized server, which is a single point of failure. They presented the ARP spoofing detection method and named it D-ARP, which is compact with an ARP table.

Nasser et al.[7] explained and performed an ARP spoofing attack. They discussed many kinds of vulnerabilities in the ARP protocol. Based on that, they found a solution for ARP spoofing attacks by developing a defensive system that will operate at the user's end to maintain the security from this kind of attack. They proposed a method to secure networks from ARP spoofing in an effective way, and they confirmed that they didn't need to add or remove new software on the device or extend ARP protocol standards to run that defensive software. It is very easy to implement in any host environment. They also think that due to its loopholes, the ARP protocol may change or become obsolete soon. Securing the users' data while communicating at both ends is still the biggest challenge for us until today. One of the types of attacks to sniff the traffic while communicating is ARP spoofing. This attack can exploit the vulnerability of the ARP protocol. It can be done just by replacing the IP and MAC of the victim. Data et al.[3] proposed a static ARP cache table solution. In this process, they are removing the process of adding the static ARP cache table manually and adding the ARP validation function to manage the static ARP cache table automatically. By this method, they can protect against all types of ARP spoofing attacks. Prabadevi et al.[9] have proposed the working principle of how ARP protocol works and they have proposed a method to mitigate

the ARP cache poisoning attack. ARP spoofing attack is one of the deadliest attacks. By using this attack, the attacker can sniff anyone's traffic.

3 Methodology

3.1 Background

An ARP spoofing attack is also known as an ARP cache poisoning attack. It is a type of Man in the Middle (MitM) attack. Through this attack, we can sniff the network traffic on the target which is communicating through the network.

3.1.1 ARP Protocol

ARP (Address Resolution Protocol) is a protocol that helps us perform network communication. All the devices in a network use IP addresses to communicate with each other, but switches identify the device with a MAC address, where we use the ARP protocol to translate the IP address to a MAC address. The ARP protocol consists of a table of mapped IP addresses to MAC addresses.

Figure 1. is an example of an ARP cache table. It can be seen by using the command "ARP-a".

3.1.2 Working Mechanism

What all of the steps do we come across in a single successful network communication? Whenever we request the network to communicate with another device, we do it using the IP address. But the problem is that switches use MAC addresses to communicate with each other. To come across this problem, we use the ARP protocol. Whenever a switch wants to know the MAC address of a device, it sends the request to all the devices with the requested IP address, and the device that matches its IP responses with its MAC address, sends it to it. The received MAC address will be stored in the ARP cache so we can use it again whenever we need it. Next time, before sending the request to all the devices to get the MAC address, it checks the ARP cache if the requested IP address is present or not to save time. This is how network communication occurs. Figure 2 shows us this whole process in a simple way.

When an ARP spoofing attack takes place, the attacker poisons the ARP cache of the router by changing the victims' MAC address to his MAC address and poisons the victims' ARP cache by changing the router MAC address to his MAC address. Now all the traffic between victim and router will be passed through the attacker's PC, so with that the attacker can sniff all the traffic of the victim. Figure 3 explains this whole ARP spoofing.

3.1.3 Types of Man-in-the-Middle Attacks

Active session attack During an active session attack, the attacker stops the victim from communicating with the server and starts communicating

```
C:\Users\Asus>arp -a

Interface: 192.168.179.1 --- 0x3
  Internet Address      Physical Address      Type
  192.168.179.254       00-50-56-f0-d2-58     dynamic
  192.168.179.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.33.240 --- 0x7
  Internet Address      Physical Address      Type
  192.168.33.117        1a-05-63-ee-10-e6     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.17.1 --- 0xf
  Internet Address      Physical Address      Type
  192.168.17.254        00-50-56-ef-7f-66     dynamic
  192.168.17.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 1: ARP cache

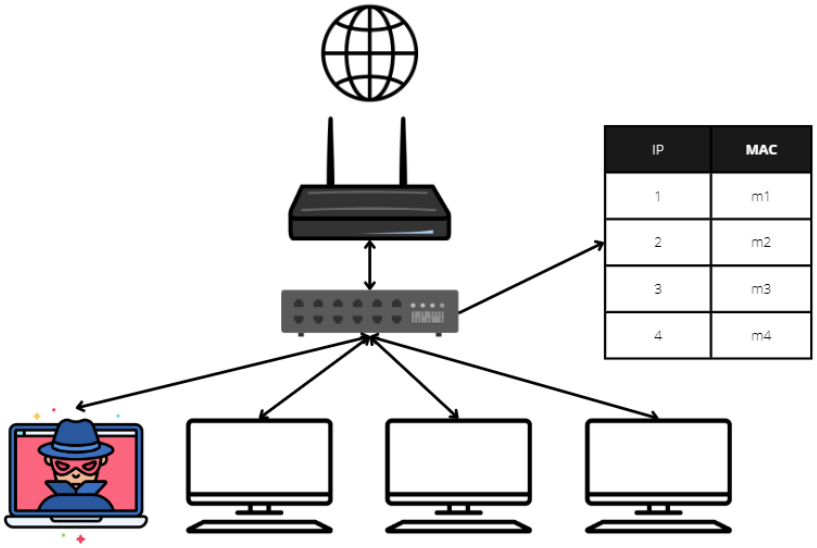

Figure 2: Before Attack

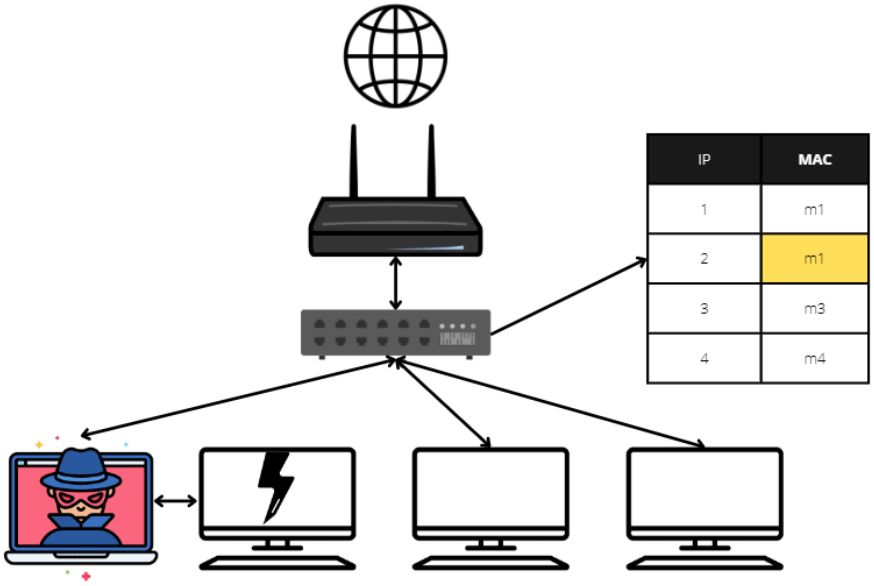

Figure 3: After Attack

with it by changing the actual data sent by the victim. In this attack, the attacker will be able to change the data communication between two points.

Ex.: Session Hijacking

Passive session attack In a passive session attack, the attacker sniffs the data which is communicating between two points and uses it for his own purpose. In this case, the attacker will not be able to change the data; he will be able to see it.

Ex.: ARP spoofing

3.1.4 Tools to perform ARP spoofing

There are many tools available on the market to perform this attack. Among them, these are the most popular tools used by people.

- Arpoison
- Cain and Abel
- KickThemOut
- Aranea
- Larp
- Netcommander
- Arpspoof
- Ettercap

In this paper, we use the Ettercap tool to perform ARP spoofing.

3.2 Problem Statement

95% of HTTPS servers are vulnerable to MITM, MITM attacks were involved in 35% of exploitations, and 10% of companies implement HSTS. [Meth1]

Many researchers have worked on developing the software to mitigate and detect ARP spoofing for SDN and have proposed solutions [2, 1, 10, 4]. Some of the researchers have developed external tools to detect the ARP spoofing [8, 5, 6]. The architecture and models of ARP spoofing detection and mitigation are predominantly available, but only a handful of researchers [7, 3, 9] have touched upon the architecture and models.

Almost all of the researchers are talking about the detection and mitigation part. Very few researchers are interested in the attack part. I am proposing the proper attacking module and method of ARP spoofing. Most researchers and pen testers use the manual method to perform ARP spoofing attacks. Many of these pen testers don't know the correct method to perform this attack to

overcome this problem. I am proposing an automated python script[Meth2] to carry out ARP spoofing attacks in a single click. To prevent these types of attacks, I am proposing a model to detect the ARP spoofing attack.

3.3 Proposed Solution

In this paper, I have proposed a basic model of an ARP spoofing attack and the detection of ARP spoofing. And to make the work easier, I have proposed a Python script for ARP spoofing. By just mentioning the target IP, we can perform the ARP spoofing attack.

3.3.1 Architecture

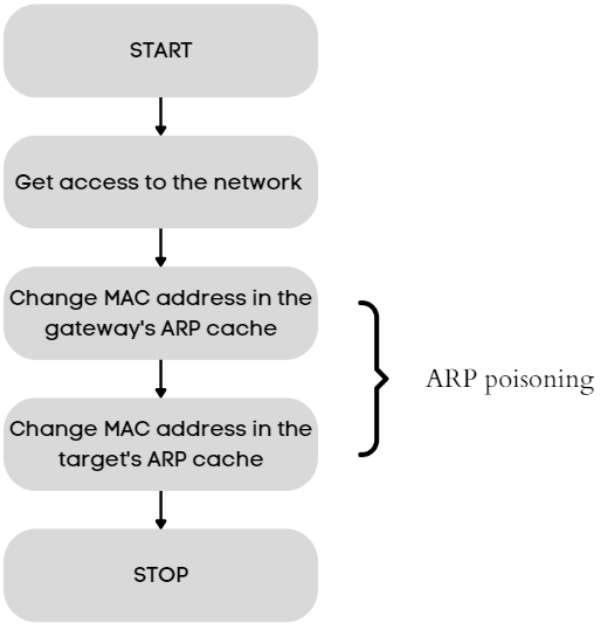

Figure 4: ARP spoofing attack model

ARP spoofing attack model While performing the ARP spoofing attack, first we have to connect to the network where the target is located. The ARP cache will consist of a MAC address for every IP address in it. In the gateways' ARP cache, we have to replace the targets' MAC address with our MAC address and vice versa in the targets' ARP cache. By doing this, we will become an intermediate between target and gateway, so that we can sniff the traffic between them.[Figure4]

ARP spoofing can be done by many tools, as mentioned in 3.1.4. The command to perform ARP spoofing by Ettercap is,

- sudo ettercap -T -S -i wlan0 -M arp:remote /gateway IP// /target IP//

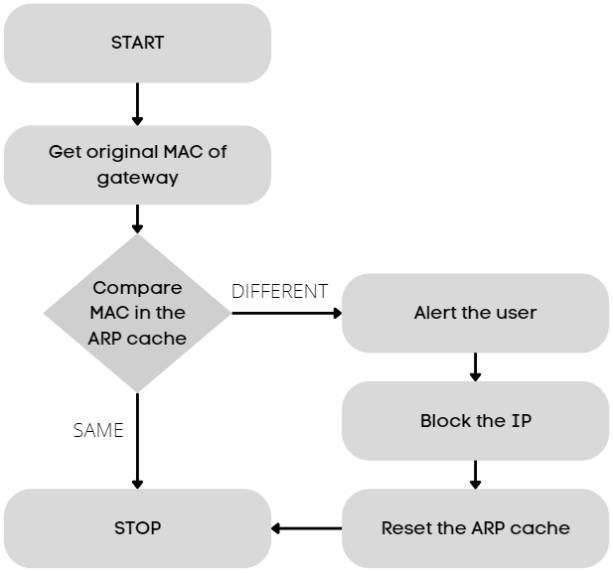

Figure 5: ARP spoofing detection model

ARP spoofing detection model To detect ARP cache, it is very simple. First, we have to know the original MAC address of gateway. Now we have to compare that MAC address in our ARP cache. If the MAC in ARP cache is same then it's fine, else it is ARP poisoning. Block that IP and reset the ARP cache. [Figure 5]

3.3.2 Implementation

Hardware and Software requirements Internet connection is required to establish the network. Laptop or PC and smart phones are required to connect the network. Kali Linux OS[Meth3] is used for attacking. NMAP[Meth4] is used for scanning the live devices. ETTERCAP[Meth5] is used to perform the ARP spoofing and WIRESHARK[Meth6] is used for traffic analysis.

Python Script Start

```
import subprocess
```

```
def info():
    print(r"""
        ____   _______     ________        __   ______   __________
       / __ \/ ____/ |   / /   _/ /       \ \/ / __ \/ ____/ _/
      / / / / /__/  | | / // // /         \  / / / / /  __ / /
     / /_/ / /___  | |/ // // /___        / / /_/ / /_/ // /
    /_____/_____/  |___/___/_____/       /_/\____/\____/___/
    """)
    print(r" ------------------------------------------------------------")
    print(r'| * Syntax: "sudo python3 ARP.py"
|')
    print(r"| * This code performs ARP spoofing attack
|")
    print(r"| * Recommended for Kali Linux
|")
    print(r"| * Author: Ygendra R Bijapur
|")
    print(r" ------------------------------------------------------------")

def ip():
    subprocess.run(["ifconfig"])


def nmap():
    ip = input("\nEnter IP Range,\nEg:198.172.14.50-200==>")
    subprocess.run(["route", "-n"])
    subprocess.run(["nmap", "-sP", ip])


def ARP_spoofing():
    t1 = input("\nEnter the gateway IP :")
    t2 = input("\nEnter the target IP from above list:")
    ani = input("\nEnter active network interfaces,
    \nEg: eth0,wlan0,l0,etc  ==> ")
    subprocess.run(["ettercap", "-T", "-S", "-i", ani, "-M", "arp:remote",
    "/"+t1+"//", "/"+t2+"//"])

info()
ip()
nmap()
ARP_spoofing()
```

Output The python script ARP.py is executed in the Kali Linux terminal with the command "sudo python3 ARP.py". And required information is provided, like IP, range, and network interface. Respective Figure 6,7,8,9

Figure 6: Interface

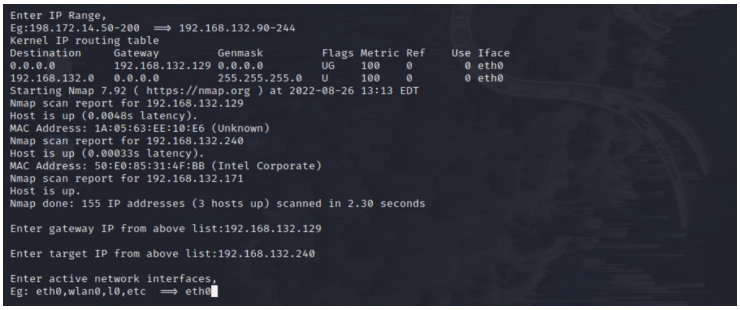

Figure 7: Interface

is the generated output after this prose. To confirm whether the attack was executed successfully, Wireshark is used with the filter "ip.addr==victim IP". Figure 10 is the output of Wireshark.

4 Results

To achieve a successful Arp spoofing attack, we must have a proper set up of the network. I have performed this using my smart phone and laptop. I have connected my laptop to my mobile with a hotspot. Here, the laptop acts as a victim, the mobile acts as a gateway, and Kali Linux acts as an attacker in a virtual machine.

ETTERCAP is used to poison the ARP cache, NMAP is used to scan the network, and WIRESHARK is used for traffic analysis.

We can access the source code of ETTERCAP using this link[Result1]. This tool contains a graphical version, so it will be easy for users. Alberto Ornaghi

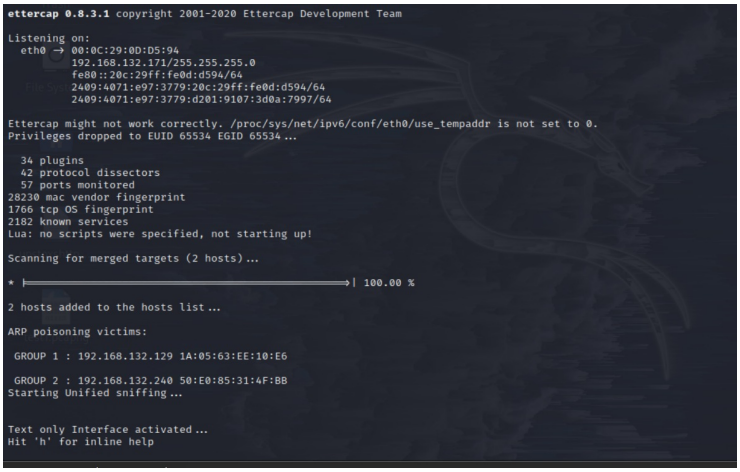

Figure 8: output

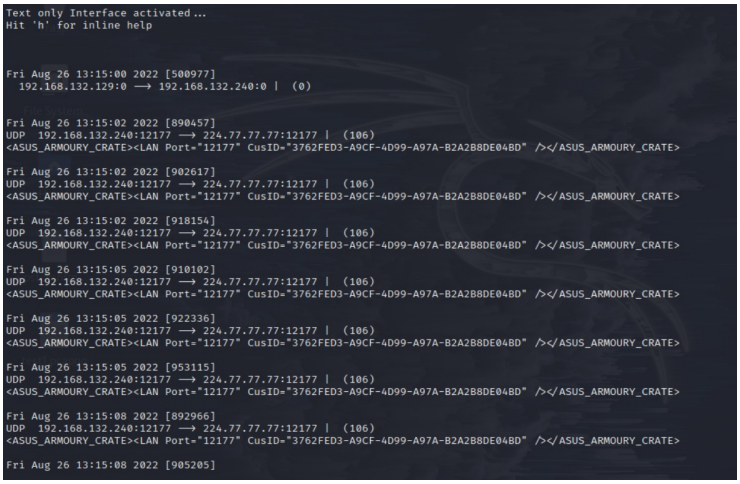

Figure 9: output

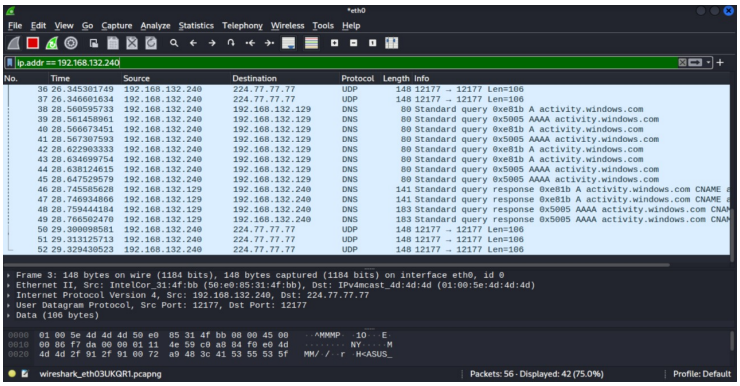

Figure 10: wire-shark

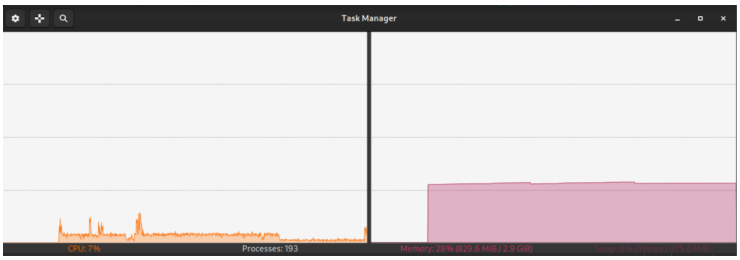

Figure 11: This is the CPU and memory usage of Kali Linux before the attack

is the original author of this tool.

Figure 11 shows the usage of CPU and memory before the attack, and Figure 12 shows the usage of CPU and memory after the attack using the proposed tool. As we can see, there is high CPU usage while attacking as compared to before the attack graph. As we can see, there is a negligible difference between the memory usage graphs. This data is collected using the task manager in Kali Linux.

5 Conclusion and Future Work

In this paper an improved model of ARP spoofing detection is proposed which will block the attackers IP so that it will be not easy for attacker to attack again and notifies the owner of network which helps us to take action on attacker. An automated python script is proposed for ARP spoofing attack which will helps us faster the attack. The researchers are also working day and night to improvise in both attacking and defending field of MITM attack. So here I am more interested is attacking part I will try to improvise my script and make it more effective.

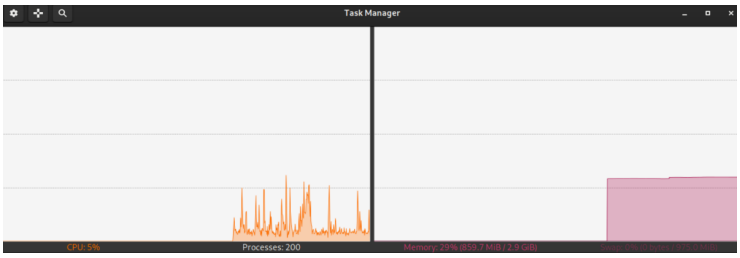

Figure 12: This is the CPU and memory usage of Kali Linux after an attack

References

[1] D. Balagopal and X. A. K. Rani. A technique for a software-defined and network-based arp spoof detection and mitigation. *Int J Appl Eng Res*, 13:14823–14826, 2018.

[2] S. Buzura, M. Lehene, B. Iancu, and V. Dadarlat. An extendable software architecture for mitigating arp spoofing-based attacks in sdn data plane layer. *Electronics*, 11(13):1965, 2022.

[3] M. Data. The defense against arp spoofing attack using semi-static arp cache table. In *2018 International conference on sustainable information engineering and technology (SIET)*, pages 206–210. IEEE, 2018.

[4] H. Y. KHALID, P. M. ISMAEL, and A. B. AL-KHALIL. Efficient mechanism for securing software defined network against arp spoofing attack. *Journal of Duhok University*, 22(1):124–131, 2019.

[5] A. Majumdar, S. Raj, and T. Subbulakshmi. Arp poisoning detection and prevention using scapy. In *Journal of Physics: Conference Series*, volume 1911, page 012022. IOP Publishing, 2021.

[6] S. M. Morsy and D. Nashat. D-arp: An efficient scheme to detect and prevent arp spoofing. *IEEE Access*, 10:49142–49153, 2022.

[7] H. I. Nasser and M. A. Hussain. Provably curb man-in-the-middle attack-based arp spoofing in a local network. *Bulletin of Electrical Engineering and Informatics*, 11(4), 2022.

[8] B. Pingle, A. Mairaj, and A. Y. Javaid. Real-world man-in-the-middle (mitm) attack implementation using open source tools for instructional use. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0192–0197. IEEE, 2018.

[9] B. Prabadevi and N. Jeyanthi. A framework to mitigate arp sniffing attacks by cache poisoning. *International Journal of Advanced Intelligence Paradigms*, 10(1-2):146–159, 2018.

[10] J. Xia, Z. Cai, G. Hu, and M. Xu. An active defense solution for arp spoofing in openflow network. *Chinese Journal of Electronics*, 28(1):172–178, 2019.

Methodology References

[Meth1] SecureOps. Critical cybersecurity statistics you must know for the last several years. https://www.secureops.com/wp-content/uploads/2021/08/SecureOps-Cybersecurity-Statistics-Report-FINALv1-updated.pdf.

[Meth2] Python Software Foundation. Python. https://www.python.org/.

[Meth3] OffSec Services Limited. Kali linux. https://www.kali.org/.

[Meth4] Gordon Lyon. Nmap. https://nmap.org/.

[Meth5] Alberto Ornaghi. Ettercap. https://www.ettercap-project.org/.

[Meth6] Gerald Combs. Wireshark. https://www.wireshark.org/.

Result References

[Result1] Alberto Ornaghi. Ettercap. https://github.com/Ettercap/ettercap.git.