National College of Ireland

Novel Technique to Detect and Defend Address Resolution Protocol Poisoning

MSc Research Project
MSc Cybersecurity

Salint George Mukalath
Student ID: 21136548

School of Computing
National College of Ireland

Supervisor: Mr. Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Salint George Mukalath……

Student ID: ……21136548……..……

Programme: ……MSc Cybersecurity…… **Year:** 2022-23

Module: ……MSc Research Project……..……

Supervisor: ……Mr Jawad Salahuddin……

Submission Due Date: …..15 Dec 2022……..……

Project Title: Novel Technique to Detect and Defend Address Resolution Protocol Spoofing

Word Count: ……6066…… **Page Count**……20……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Salint George Mukalath……

Date: …15 Dec 2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Novel Technique to Detect and Defend Address Resolution Protocol Poisoning

Salint George Mukalath
21136548

Abstract

The world of today is now completely dependent on networks and protocols designed to make setting up these networks easier, however some of them did not take security safeguards. Usually, security team not giving more importance in the security of layer 2 of the Open System Interconnect (OSI) layers, which is the main motive of this project. Since there is no authentication mechanism provided by ARP for inbound request packets, any user may use fake ARP message and insert malicious data to corrupt the target machine's ARP cache. ARP spoofing is used by attackers for interfering in the communication between two devices and performing a man in the middle (MITM) attack. MAC and IP addresses of the packets of data being transferred in a network will analyzed by the system for performing ARP spoofing attack detection. The system will alert the user on detecting the attack and it will restore the ARP cache table to its original form as the table will be modified when the attack occurs. The scapy library will be used for obtaining the MAC and IP addresses and the performance of the system developed will be evaluated by generating an attack on a device in which the detection system is running. The result of the approach shows that the ARP spoofing attack can be effectively detected based on the analysis of the MAC and IP addresses of the data packets.

1 Introduction

The internet, which serves as an effective and dependable network for its billions of users, is essential for maintaining the communication. The requirement to secure the data transmitted across the network is increasing day by day. Content transmitted via internet is sent as distinct packets that travel over various routes in a certain order over time before reuniting at the ultimate end node. Nowadays all smartphones and computers are connected to some type of networks, these devices might be connected to an Internet Service Provider (ISP) directly or by a Local Area Network (LAN) along with other devices.

The rules of communication protocols are abused by hackers in order to interfere with device communication and gain access to sensitive information. This sensitive information being exposed may lead to a bunch of problems for individuals as well as organizations who might have been the target of these attackers. Low level layer assaults are one of the biggest dangers in the online platform.

One protocol being used for communication in networks is the Address Resolution Protocol (ARP) which is responsible for mapping logical addresses or Internet Protocol (IP) address to physical address or Media Access Control (MAC) address on a network (Plummer, 1982). Before the network layer encapsulated in a data link layer frame, the host should be aware of the MAC address of the recipient before the data packets sent. For finding a particular IP address in a network the ARP request packet is broadcasted by the source node of the host to determine the owner of the specific IP address. The request packet is acknowledged by all nodes in the LAN and the node which has the IP address that is being requested sends its MAC address as reply. The reply from the node will be sent only to the source that sent the request. This reply will be received by the host and the received MAC and IP address will be saved in an ARP table. This table is used as cache and a request will only be sent by the node of the host only if the ARP table is empty.

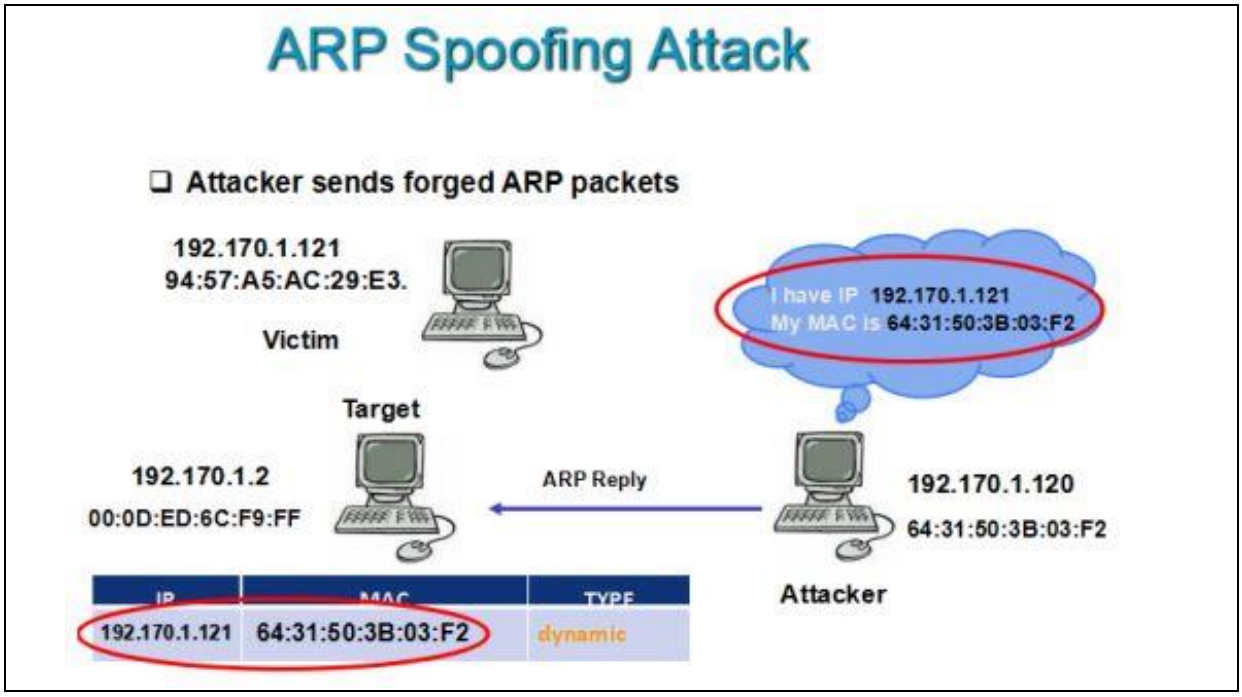

Figure 1: ARP spoofing attack (Bijral, Gupta and Sharma, 2017)

The main weakness of the ARP protocol is that it is a stateless protocol which means that machine receives an ARP reply even if there is no request has been sent by it, this vulnerability is exploited by attackers using the ARP spoofing attack in which ARP reply messages containing the IP address of a network resource like the DNS server or gateway are sent to a machine that is targeted by the attacker. The MAC address of the network resource will be replaced with the MAC address of the attacker's machine (Figure 1). The device receiving the replies will not be able to identify if it is authentic and the ARP table will contain the MAC address of the last ARP reply. The attacker will then become man in the

middle and any data sent to the authentic source will always pass through the machine of the attacker hence the attacker is able to read and modify the data. The end user unaware of the attack since the attack take place in the lower-level TCP/IP protocol. Denial of service (DoS) is another type of attack through ARP spoofing as the attackers drop the packets being sent and this will result in service being denied to the machine which is targeted by the attacker or the victim machine with tools for creating ARP spoofing attacks like Ettercap which is relatively easy for carrying out a spoofing attack (Sukkar *et al.*, 2016). Hence securing machines against ARP spoofing a detection mechanism is required.

1.1 Research Question

- How will the ARP spoofing attacks occurring in a network be detected?
- How will the effectiveness of the ARP spoofing detection and defence method be demonstrated?

1.2 Aims

- Create a technique for detecting ARP spoofing in a network.
- Update the ARP table after detecting the spoof attack for defence.
- Showcase the effectiveness of the technique used for detecting the ARP spoofing attacks.
- Contributing the knowledge to layer 2 security attacks.

2 Related Work- Literature review

2.1 ARP Spoofing and MITM detection

A framework detecting a rouge user impersonating another user in a network was proposed in (Bhattacharya *et al.*, 2022). Security measures were present in the framework for preventing attacks on itself and for making that a single point of failure does not exist. RSA was used as the security mechanism in the framework. Random time intervals were initialized in the framework for making sure client messages were not sent by other devices. The AES encryption was used for all the messages that were sent from the framework. The main limitation of the approach is that the framework proposed here only detects DoS attacks and there is no complete defense.

The conditions for carrying out MITM using ARP spoofing was analyzed in(Stepanov *et al.*, 2021). This approach showed how ARP spoofing attacks can be generated by using the scapy library in Python programming language. Some examples of MITM attacks are like ICMP redirection and DHCP spoofing also showcased in this approach. The main findings from this approach reveal that the threats due to the interception of traffic in a network are a grave threat to the security of a network. The effectiveness of the scapy library in generating attacks was also exhibited in this approach. This approach focused only on the detection of attacks like ICMP redirection and DHCP spoofing and MAC spoofing while no other kinds of attacks were detected in this approach.

ARP poisoning or spoofing attacks were detected using an algorithm in (Majumdar, Raj and Subbulakshmi, 2021). The algorithm analysed the authentic MAC address and the response MAC address of an ARP packet. The scapy library was used for the implementation of both the ARP poisoning attack detection and prevention algorithms. From the results of the approach, it was observed that scapy library is effective in detecting ARP poisoning attacks. The main limitation associated with this approach is that the algorithm that was developed here was not able to detect any attacks other than ARP poisoning attacks.

The working of the ARP spoofing attacks and a number of existing methods that provided security against the ARP spoofing attacks was studied in (Hijazi and Obaidat, 2019). The results of the study proclaim that both the prevention and detection systems should be implemented with consideration for minimizing the cryptographic methods in the network. The study also defined four security requirements needed in the development of mechanisms for detecting and preventing ARP spoofing attacks which are, 1) the cryptographic process has to be reduced in the mechanism, 2) the detection mechanism has to be easily applicable, universal and consistent with ARP, 3) acknowledge all the ARP attacks, 4) acknowledge the cost of network management. The drawback of this approach is that no method or technique for detecting ARP spoofing attacks were developed in this approach based on the studies and guidelines discussed.

(Rohatgi and Goyal, 2020) discuss about ARP spoofing attacks and the benefits and disadvantages of a number of ARP spoofing detection techniques were compared and discussed. The main limitation of this approach was that no methods that were studied in this approach were implemented for observing the effectiveness of the detection methods.

Three methods were proposed for the detection and prevention of ARP spoofing in the paper of (Agrawal, 2019). The first method used static IP addresses, the second suggested the development of a new protocol and the third one proposed the use of tools or software. The results of the approach revealed that the first method was the most effective among the three while the third was found to be expensive and difficult to execute; but it lacks with no detection and defense of the spoofing attacks.

Different tools and methods for detecting preventing ARP spoofing attacks were proposed in (Assegie and Nair, 2005). The ARP spoof tool was used in this approach for generating an ARP spoof attack on a machine to study how the ARP cache table is maintained by the host using the spoofed MAC. The different ARP spoofing attack detection tools like Wireshark, ARP watch, XARP and ARP Alert were compared and evaluated in this approach. This approach only highlights how the last MAC address is stored in the ARP table as the machine is not able to differentiate between the MAC of an authentic device and the device of an attacker.

Traditional network-based ARP spoofing attack was studied in (X. Jing et al ,2019). The threat level of the ARP network was determined by performing ARP spoofing in a Software Defined Networking (SDN) network. A unique method based on the OpenFlow platform was

suggested in this approach for detecting ARP spoofing attacks. The mechanism was analysed and implemented on the POX controller as a module. The mechanism was observed to reduce the threat of ARP spoofing in SDN platforms in OpenFlow platforms.

Various methods for mitigating ARP cache poisoning attack in SDN were discussed in (Shah and Cosgrove, 2019). The methods are mainly divided into three groups which were IP-MAC Address Bindings based solutions, Traffic Patterns based solutions and Flow Graph based solutions. All the methods in the three groups were evaluated based on the working principles, advantages, and disadvantages; but there is no relevant information for the defense of ARP spoofing as it is based only on SDN.

The ARP poisoning attacks and different mechanisms for detecting attacks were studied in (Divya and Christopher, 2019)The methods for detecting the attacks were compared and evaluated in this approach and the result of this approach reveals the techniques that can be used for creating a framework which can be used for ARP poisoning attack mitigation and detection . Although the approach analyses the techniques that can be used for creating a ARP poisoning attack detection framework, no such framework was implemented in this approach.

(Jeong, Kim and Jo, 2022) advise an access point-based ARP Spoofing Detector (ASD) for detecting ARP spoofing attacks. The ASD was able to detect the ARP spoofing attacks without a false positive rate being returned. The ASD was able to distinguish between VM connections and ARP spoofing attacks by utilising three information tables DHCP table, ARP cache table and AssocList. These tables are maintained commonly by the access points based on the Linux system. The evaluation of the ASD used in this approach reveals that it was able to effectively detect ARP spoofing attacks without a false positive rate. The significance of the ARP cache table in the detection of ARP spoofing attacks was revealed from the results of this approach which is not enough to defend the complete defense of spoofing.

(Bijral, Gupta and Sharma, 2017) discusses the vulnerabilities of the ARP the attacks like session hijacking and cookie hijacking are explained in this approach. The Snort was used in this approach for detecting the ARP spoofing attacks. The performance of the Snort was evaluated in this approach; it is observed that the Snort was able to detect the MAC address of the intruder in a network, but the efficiency level is too limited.

A deep learning model was used for the detection of an ARP spoofing attack in (JapneetKaur, 2022). A hybrid Convolutional Long Short-Term Memory (ConvLSTM) was used in this approach. The hybrid ConvLSTM-ECC model performs prediction features extracted from the raw data; these features are used by the model for detecting the MITM attacks based on ARP spoofing in the data link layers. The detection is performed by the LSTM while the feature extraction is performed by the convolutional layers. From the results of this approach, the deep learning model can detect ARP spoofing attacks. The main limitation of the approach is that it was able to detect only MITM attacks based on ARP spoofing but there is no efficient mitigation.

(Abdulla, Al-Raweshidy and S. Awad, 2020) reveals an artificial intelligence-based method for detecting ARP spoofing attacks in IoT networks with integration of Neural networks. The dataset used for training the neural networks contained the data associated with packets of different protocols like the TCP, UDP and ARP. The neural network can be able to achieve an accuracy of more than 90% in detecting ARP spoofing attacks. This paper is only focusing on IoT networks and not giving full spoofing mitigation in the data link layer.

An approach containing 3 modules used for the detection and mitigation of ARP spoofing in (Galal, Ghalwash and Nasr, 2022) The first module is to provide permission for the first time and store data in the database. The MD5 was used here for security. Module 2 is used for making sure the exclusion of internal ARP, lastly Module 3 is for analysing if a single IP address has 2 MAC's or if a single MAC has 2 IP addresses. There is a database for storing IP address values. The results of the approach revealed that it can detect ARP spoofing attacks in a network. The main downside of this approach is that it was not able to detect attacks other than MITM and DoS.

2.2 Research Niche

Table 1: Literature comparison

Related Work	Strength	Limitation
(Rohatgi and Goyal, 2020)	Manually bind MAC and IP in the ARP table to avoid spoofing.	Implementation is a hectic job in large network.
(Ibrahim *et al.*, 2020)	A server stores MAC & IP data with encryption.	Continuous update of MAC and ARP, low speed broadcast
(Sakhawat *et al.*, 2019)	AACPD is used for the spoof detection	Low accuracy and more workload
(Ahn, Lee and Kim, 2019)	SCADA secretly monitor the intrusion and alter the MAC before the execution of attack	Weak functionality and low accuracy
(Zhao, Guo and Lv, 2020)	Use maths calculation and EMR-ARP to detect intrusion; keeps long term IP-MAC binding.	It should improve with error validation due to the concept is based on puzzles.
(Morsy and Nashat, 2022)	D ARP checks the authorisation of signed packets. A log file saves all track records	High memory use.
(Sun *et al.*, 2020)	Use ICMP echo packets to decide validity of the packets for the authenticity of the IP-MAC pair.	Not suitable for skilled attackers.
(Abdulla, Al-Raweshidy and S. Awad, 2020)	CBNA-RF concept detect the attacker	Finds the spoofing with reduced performance.
(Gao *et al.*, 2018)	Algorithm based static ARP	Less efficient in large

	update	networks.
(Galal, Ghalwash and Nasr, 2022)	Encrypted MAC and IP storage in a database using hash algorithm.	Not suitable for large networks. Detect only MITM and DoS

2.3 Summary

It is observed that analysing the IP or MAC address is an effective way to detect ARP spoofing attacks, moreover the literature that the performance of the methods used for detecting ARP spoofing attacks can be evaluated based on a simulated attack on a machine. The scapy library is one of the major effective for performing ARP spoofing attack detection in several approaches as the library was used for analysing the MAC or IP addresses and for generating an ARP spoofing attack for evaluating the effectiveness of ARP spoofing attack detection method.

There is no significant research gap between the existing literature and the ARP spoof detection method proposed here. Analysing the MAC addresses using scapy library was observed to be an effective way of identifying a MITM ARP spoofing attack, moreover it uses SHA 256 algorithm for the encryption of MAC to enabling advance security. So, the method proposed ensures that the findings of the existing literature are true based on the results of the ARP spoofing attack detection method proposed above.

3 Research Methodology

Typically, there are two methods for ARP defense in the cybersecurity field, passive system, and active system. Passive system monitors the ARP traffic in a continuous manner to find the mapping irregularities and pass the messages to the administrator. In contrast the active system constantly checks weak points of the network traffic and provides either a healing process or pass the necessary data to the concerned member of the defense system. The invented system is managed to detect MITM ARP spoofing attack. This was achieved using the methods in the scapy library. The ARP spoofing attack was made to run on a machine (this machine will be referred to as victim machine henceforth). The victim machine received data from a router (this device will be referred to as authentic device henceforth) in a normal network. When an ARP spoofing based MITM attack takes place the attacker's device (this device will be referred to as attacking device henceforth) will interfere in the communication between the victim machine and the authentic device.

3.1 ARP Spoofing detection

The ARP spoofing attack detection system used the methods in the scapy library for obtaining the MAC address and IP address of a packet of data that was sent to the victim

machine. The MAC address of the authentic device was saved by the system and when the victim machine received a packet of data, the MAC address of the particular packet was extracted and compared with the previously saved MAC address of the authentic device. This will be performed for every packet of data sent to the victim machine.

Now when an attacking device comes in between the victim machine and authentic device it performs a MITM attack by ARP spoofing. The attacking device which was now between the victim machine and authentic device will send data packets to the victim machine acting as the authentic device. So authentic device will consider the attacking device as the victim machine and will send data to it unknowingly and the attacking device will gain access to the data shared. The victim machine will also send the data to the attacking device unknowingly considering it as the authentic device.

The ARP cache table in the victim device will store the details of the MAC address and IP address of the authentic device, but when the MITM attack occurs the MAC address of the authentic device in the ARP cache table will change to the MAC address of the attacking device, as the attacking device comes in between the victim machine and the authentic device, with the IP address of the authentic device remaining the same. The ARP spoofing attack detection system on detecting the attack will change the MAC address of the authentic system in the ARP cache table to the original MAC address of the authentic system which was saved earlier.

3.2 ARP Spoofing Defense

The summarization of defense mechanism is as follows,

i. Attack charged and router and host get poisoned.
ii. Start the defense system in the victim machine and it will send the ARP request packet to the router to gain the original IP and MAC address.
iii. The system saves the data with all mapping and do check the MAC address in regular time intervals and cross check with the previously saved data.
iv. The attacker starts sending poisoned ARP data and the router as well as the victim machine updates the same.
v. Instantly the defense system does the comparison of the MAC address with the previously saved data and identifies the change of the gateway MAC address.
vi. Now the MAC address of the authentic device is encrypted into a hash form by using the SHA-256 algorithm. The hash value of the MAC address of the authentic device saves by the ARP spoofing attack detection system and when the victim machine receives a packet of data, the MAC address of the particular packet extract, encrypt into a hash form using the SHA-256 algorithm. This hash value is then compared to the previously saved hash value of the MAC address of the authentic device. This is performed in every packet of data sent to the victim machine.

vii. Subsequently the program informs the victim about the attack, corrects the ARP table with original MAC/IP and completes the healing process by sending the correct IP/MAC to the router.

viii. Both the attacker and the victim's program try to alter the mapping and finally the victim's machine wins and reports the attack details.

ix. In conclusion the victim's machine and the gateway are safe and the attacker is unable find the network traffic.

3.3 Performance Evaluation

Initially the communication took place between the victim machine and the authentic device sent data packets to the victim machine. Now the attack was generated in the same computer in which the victim machine operated. The victim machine ran on one Operating system (OS) and the attacking device ran on another OS in the same computer. The ARP spoofing attack was generated by obtaining the MAC addresses of both the victim machine and the authentic device. For obtaining the MAC addresses to generate an attack the IP addresses of the victim machine and the gateway or authentic device were required; using both these IP addresses the MAC addresses of the victim machine and the authentic device was obtained. Once the MAC addresses are obtained the data packets are being sent to the victim machine, thus an ARP spoofing based MITM attack was carried out in the victim machine.

4 Design Specification

The proposed system is a MITM attack detection and prevention program which can be installed on each machine of the network. It informs the user when the system has a ARP spoofing attack with warning messages. Secondly it corrects the spoofed table with correct values. An administrator can easily identify the attacker machine and take necessary steps to disconnect that culprit machine from the network to avoid further attacks since the IP and MAC details can be seen in the script.

Assuming the network is safe initially the system saves all the current values such as MAC and IP of each machine and the gateway address. Further it monitors address resolution protocol (ARP) table for any changes which is not legitimate in the network caused by a man in the middle attack specifically with ARP spoofing.

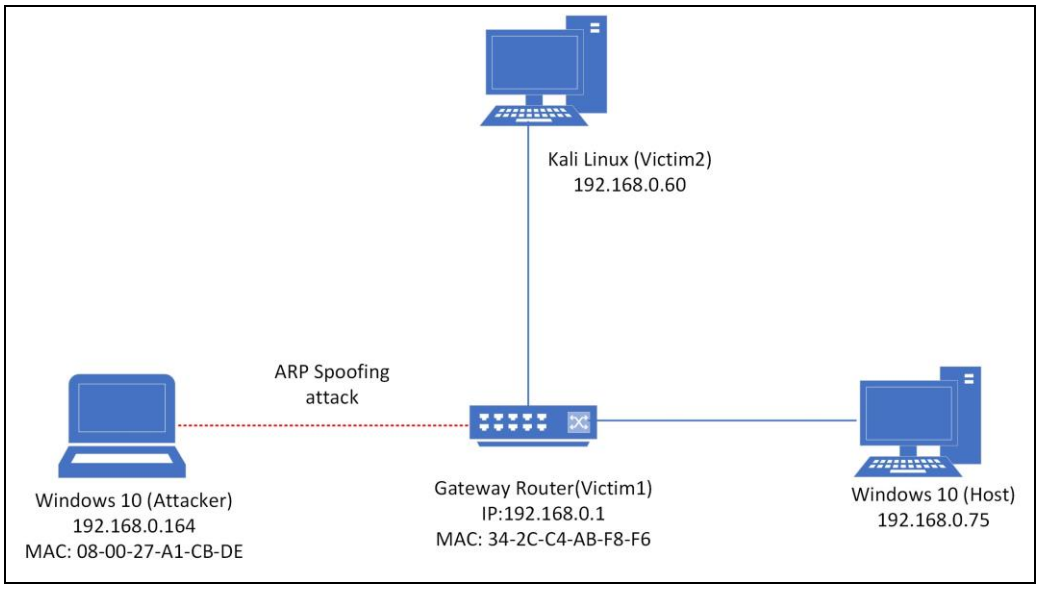

Figure 2: Network design

By using Reverse Address Resolution protocol (RARP) attacker's IP can be revealed from resolving the MAC address. After the attack happened immediately mitigation process starts; initially it reverses the MITM attack process, it can be done by correcting the ARP table in the previous state. Secondly a ping packet (ICMP) sent to the gateway device is usually router or switch forcefully by operating system to reacquire the original MAC address of gateway.

4.1 SHA-256 Algorithm

The SHA-256 algorithm was uses in this approach for encrypting the MAC addresses. The SHA-256 creates an almost unique signature for the MAC addresses (Gowthaman and Manickam, 2015). These signatures or hash values are 256-bit long and the SHA-256 algorithm can provide a high level of security to the data. For decrypting a piece of information encrypted by the SHA-256 algorithm a brute force attack would have to carry out for figuring out the encrypted information. However because of the unique nature of the encryption by the SHA-256 algorithm the hash value generated by the algorithm for two different pieces of information will never be the same (N-able, 2019).

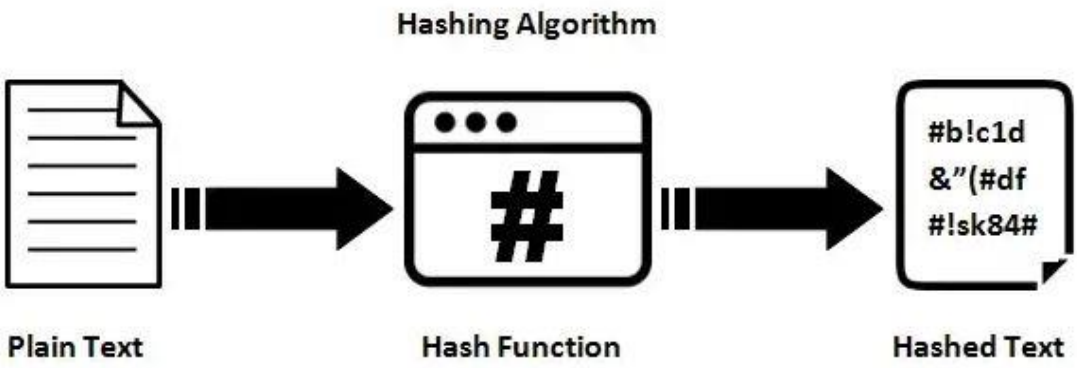

Figure 3: SHA Algorithm (Anand, 2020)

4.2 Overall Architecture

The overall architecture of the system consists of a victim machine and attacking device or machine. The ARP spoofing attack detection system obtained the MAC address from the packets of data sent by the attack device and these addresses were compared with the hash values of the MAC addresses of the authentic device and if the hash values were different the ARP spoofing attack was detected by the detection system. On detection the system alerted the user and updated the ARP table (Figure (5)).

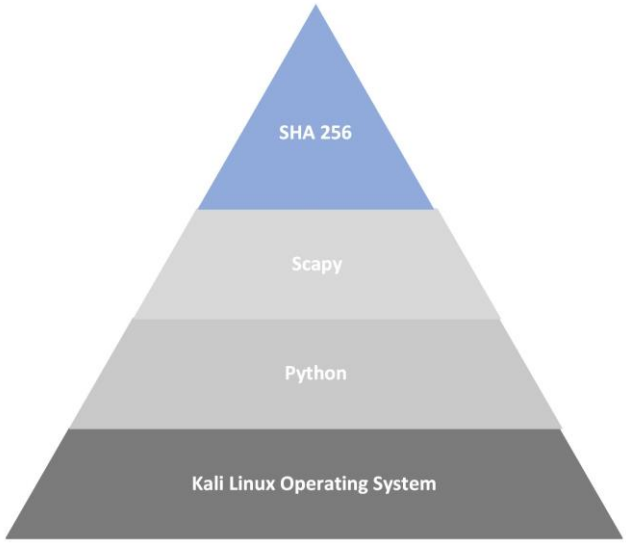

Figure 4: Defense Architecture

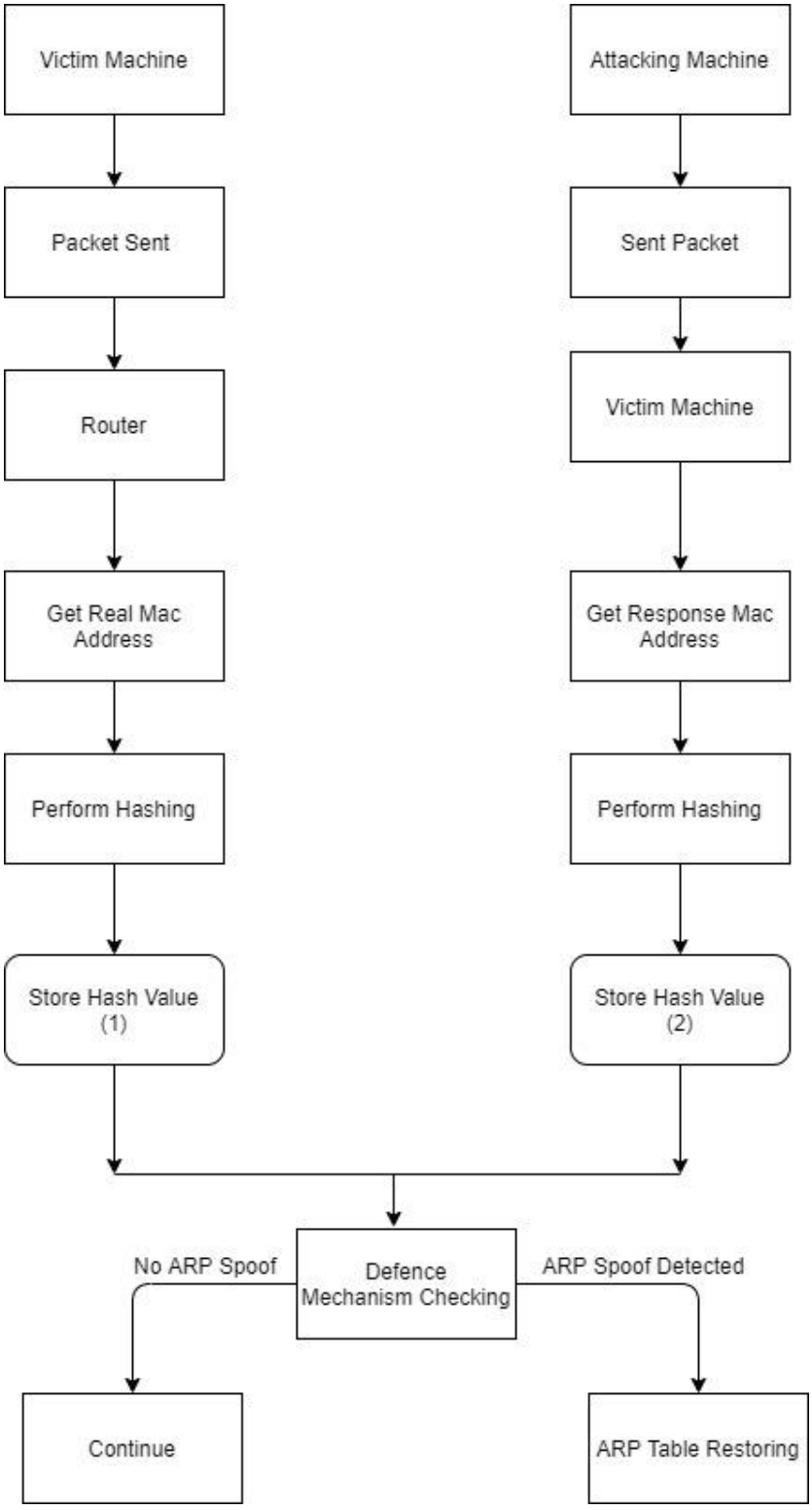

Figure 5: Architecture of the system

5 Implementation

The aim is to set up two efficient easy to use tools; one is for attacking with MITM techniques and the other is for defense from MITM attacks which should be easy to handle by a normal user without much technical knowledge.

- Because of the abundance of network-related commands and compatibility with network programming frameworks, Linux based computer systems are selected as the development kit. Kali Linux with Debian repository is a more stable security tools and standard defense system.
- Python is the development language as it is open source and compatible with multiple platforms and performs with simple and complex development. Version 2.7.18 is used for the testing and it proves the result for the successful execution which also firmly work with scapy library for this development.
- Scapylibrary contains different classes providing the method for sniffing and manipulating the data packets for a particular purpose which we defined in this research paper.
- SHA 256 hashing algorithm is used for data integrity validation based on cryptography. It has 256 bits in size.
- Anaconda framework is for the environment set up for the attacking module.

The MAC and IP addresses were obtained from the data packets using the methods in the scapy library and this was implemented using the Python programming language. For the attack simulation the victim machine and the attacking device were set up on a single computing device. The victim machine was set up on the Kali Linux OS of a computer and the attacking device was set up on the Windows 10 OS of the same computer.

The ARP spoofing attack detection system generated the results along with the alert, on the occurrence of an ARP spoofing attack in the command prompt window in the victim machine. The ARP spoofing attack detection system on detecting the attack displayed the alert '[Danger]: You are under ARP Spoofing Attack' along with the MAC address of the attacking device. It was also specified that the ARP table is being restored.

Figure 6: Detection and healing

6 Evaluation

6.1 Experiment / Case Study

From the results of the generated ARP spoofing attack on the victim machine it was observed that the ARP spoofing detection system which ran on the victim machine was able to detect the ARP spoofing attack.

On observing the ARP table before the attack, it can be seen that the MAC address of the authentic device was ‘34:2c:c4:ab:f8:f6’ with IP address ‘192.168.0.1’ and the MAC address of the attacker was also displayed in the ARP table as ‘08:00:27:a1:cb:de’ with the IP address ‘192.168.0.164’ (Figure (4)).

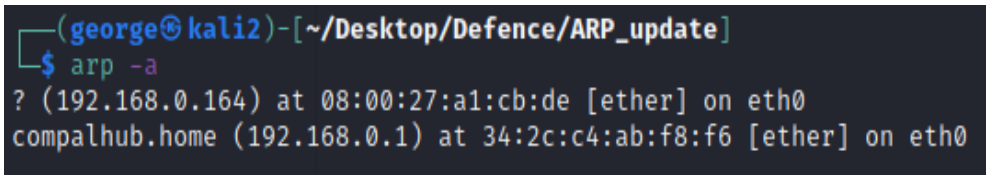

Figure 7:The ARP table before ARP Spoofing attack

During the ARP spoofing attack, it was seen from the ARP table that the MAC address of the authentic device was converted to ‘08:00:27:a1:cb:de’, which was the MAC address of the attacking device. After the ARP spoofing attack on the victim machine, it was observed from the ARP table that the two devices had the same MAC address ‘08:00:27:a1:cb:de’ but both devices had different IP addresses (Figure (5)).

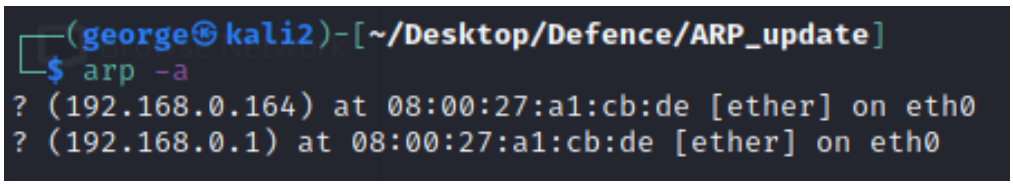

Figure 8: ARP table view during the ARP spoofing attack

The ARP spoofing attack detection system successfully detected the ARP spoofing based MITM attack and alerted the user. After the detection, the ARP spoofing attack detection system updated the ARP table so that the MAC address of the authentic device was restored to the original MAC address ‘34:2c:c4:ab:f8:f6’ (Figure (6)).

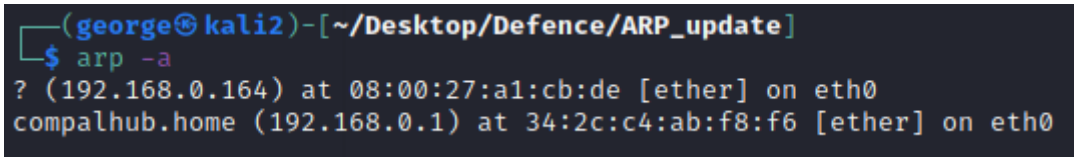

Figure 9: ARP table after restoration

6.2 Discussion

The ARP spoofing attack detection system developed here was able to successfully detect the ARP spoofing attack on the victim machine and alert the user. The ARP table was also restored to its original form after the attack. The method for ARP spoofing attack detection developed here could be compared to some of the existing literature studied. It was seen that the results of the system developed here supported the findings from Stepanov (2021) and Majumdar, Raj and Subbulakshmi (2021) as it was observed from this literature that the scapy library was effective in analysing MAC or IP addresses and performing ARP spoofing attack detection. It was observed by that Assegie and Nair (2019) a computer could not differentiate between the changed MAC addresses of the attacking device and authentic device, this issue was addressed in the system developed here as the ARP spoofing detection system is able to detect the changed MAC address and perform the detection. The findings from Galal, Ghalwash and Nasr(2022) and Saritakumar , Anusuya and Ajitha (2021) revealed that the analysis of the MAC address can be performed for detecting the ARP spoofing attack these findings were supported by the results obtained from the system developed here as ARP spoofing attack detection system developed here was able to successfully perform ARP spoofing detection by analysing the MAC addresses of the devices connected to the network . The findings observed from Stepanov (2021) that the scapy library could be used for creating an ARP spoofing attack was also supported by the findings from the approach proposed here as scapy was used for successfully generating an attack for testing the effectiveness of the ARP spoofing attack detection.

The system developed here has some limitations associated with it even though it was able to achieve its aim successfully. One limitation is that the system developed here did not bring anything new to the table as ARP spoofing attack detection by the analysis of MAC and IP addresses was successfully performed by a number of existing systems. The second limitation is that the system detects only the MITM attacks but the attacker performing ARP spoofing will also be able to perform DoS attacks and the system is not able to detect these kinds of attacks. The third limitation is that the performance of the system developed here was not evaluated based on any metrics that produced results as numerical values.

7 Conclusion and Future Work

A system for detecting the ARP spoofing attack was successfully developed here. The first research question has been successfully answered as the ARP spoofing attack is detected by analysing the MAC addresses of the authentic device and attacking device, which performed

attacks on a victim machine. The second research question was also answered as the effectiveness of the ARP spoofing detection system developed here was evaluated by carrying out an ARP spoofing attack on the victim machine in which the ARP spoofing attack detection system was running, the system was able to successfully detect the attack and restore the ARP table to the state it was in before the occurrence of the ARP spoofing attack. The analysis of the MAC and IP addresses and the generation of the ARP spoofing attack was performed using the methods in the scapy library.

The system could be enhanced in the future so that it could be able to detect more kinds of attacks other than the MITM attacks on a victim machine. Other techniques like machine learning could be used in the future for improving the attack detection performance of the system.

References

Abdulla, H., Al-Raweshidy, H. and S. Awad, W. (2020) 'ARP Spoofing Detection for IoT Networks Using Neural Networks'. Rochester, NY. Available at: https://doi.org/10.2139/ssrn.3659129.

Agrawal, G. (2019) *Detection and Prevention of ARP-Spoofing Attacks*.

Ahn, S., Lee, T. and Kim, K. (2019) 'A Study on Improving Security of ICS through Honeypot and ARP Spoofing', in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 964–967. Available at: https://doi.org/10.1109/ICTC46691.2019.8939925.

Anand, A. (2020) *Breaking Down : SHA-256 Algorithm*, *Medium*. Available at: https://infosecwriteups.com/breaking-down-sha-256-algorithm-2ce61d86f7a3 (Accessed: 15 December 2022).

Assegie, T.A. and Nair, P.S. (2005) 'COMPARATIVE STUDY ON METHODS USED IN PREVENTION AND DETECTION AGAINST ADRESS RESOLUTION PROTOCOL SPOOFING ATTACK', . *Vol.*, (16), p. 11.

Bhattacharya, D. *et al.* (2022) 'DetecSec: A Framework to Detect and Mitigate ARP Cache Poisoning Attacks', in V. Suma et al. (eds) *Evolutionary Computing and Mobile Sustainable Networks*. Singapore: Springer (Lecture Notes on Data Engineering and Communications Technologies), pp. 997–1007. Available at: https://doi.org/10.1007/978-981-16-9605-3_70.

Bijral, R., Gupta, A. and Sharma, L. (2017) 'Study of Vulnerabilities of ARP Spoofing and its detection using SNORT', *International Journal of Advanced Computer Research*, 8. Available at: https://doi.org/10.26483/ijarcs.v8i5.4016.

Divya, C. and Christopher, D.F.X. (2019) 'Security against ARP Spoofing Attacks using Bayesian Support Vector Regression', 8(7).

Galal, A.A., Ghalwash, A.Z. and Nasr, M. (2022) 'A New Approach for Detecting and Mitigating Address Resolution Protocol (ARP) Poisoning', *International Journal of Advanced Computer Science and Applications (IJACSA)*, 13(6). Available at: https://doi.org/10.14569/IJACSA.2022.0130647.

Gao, W. *et al.* (2018) 'ARP Poisoning Prevention in Internet of Things', in *2018 9th International Conference on Information Technology in Medicine and Education (ITME). 2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 733–736. Available at: https://doi.org/10.1109/ITME.2018.00166.

Gowthaman, A. and Manickam, S. (2015) 'Performance study of enhanced SHA-256 algorithm', 10, pp. 10921–10932.

Hijazi, S. and Obaidat, M.S. (2019) 'Address resolution protocol spoofing attacks and security approaches: A survey', *SECURITY AND PRIVACY*, 2(1), p. e49. Available at: https://doi.org/10.1002/spy2.49.

Ibrahim, H.Y. *et al.* (2020) 'A Secure Mechanism to Prevent ARP Spoofing and ARP Broadcasting in SDN', in *2020 International Conference on Computer Science and Software Engineering (CSASE). 2020 International Conference on Computer Science and Software Engineering (CSASE)*, pp. 13–19. Available at: https://doi.org/10.1109/CSASE48920.2020.9142092.

Jeong, Y., Kim, H. and Jo, H.J. (2022) 'ASD: ARP Spoofing Detector Using OpenWrt', *Security and Communication Networks*, 2022, p. e2196998. Available at: https://doi.org/10.1155/2022/2196998.

Morsy, S.M. and Nashat, D. (2022) 'D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing', *IEEE Access*, 10, pp. 49142–49153. Available at: https://doi.org/10.1109/ACCESS.2022.3172329.

N-able (2019) *SHA-256 Algorithm, N-able*. Available at: https://www.n-able.com/blog/sha-256-encryption (Accessed: 13 December 2022).

Plummer, D.C. (1982) *RFC0826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*. USA: RFC Editor.

Rohatgi, V. and Goyal, S. (2020) 'A Detailed Survey for Detection and Mitigation Techniques against ARP Spoofing', in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 352–356. Available at: https://doi.org/10.1109/I-SMAC49090.2020.9243604.

Sakhawat, D. *et al.* (2019) 'Agent-based ARP cache poisoning detection in switched LAN environments', *IET Networks*, 8(1), pp. 67–73. Available at: https://doi.org/10.1049/iet-net.2018.5084.

Shah, Z. and Cosgrove, S. (2019) 'Mitigating ARP Cache Poisoning Attack in Software-Defined Networking (SDN): A Survey', *Electronics*, 8(10), p. 1095. Available at: https://doi.org/10.3390/electronics8101095.

Stepanov, P.P. *et al.* (2021) ‘The problem of security address resolution protocol’, *Journal of Physics: Conference Series*, 1791(1), p. 012061. Available at: https://doi.org/10.1088/1742-6596/1791/1/012061.

Sukkar, G.A. *et al.* (2016) ‘Address Resolution Protocol (ARP): Spoofing Attack and Proposed Defense’, *Communications and Network*, 8(3), pp. 118–130. Available at: https://doi.org/10.4236/cn.2016.83012.

Sun, S. *et al.* (2020) ‘Detecting and Mitigating ARP Attacks in SDN-Based Cloud Environment’, in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 659–664. Available at: https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162965.

Zhao, Y., Guo, R. and Lv, P. (2020) ‘ARP Spoofing Analysis and Prevention’, in *2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA)*. *2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pp. 572–575. Available at: https://doi.org/10.1109/ICSGEA51094.2020.00130.