CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Network security provides the necessities and policies given by the network administrator to avoid and control the unauthorized usage, maltreatment and alteration of data. Network security allows network administrator to check the authorization of using data in the network. Each and every user has their own user id and password for accessing the data with their authority in the network. Generally network security has used both private and public computer networks. These services are used in our everyday scenarios. Networks can either be a private or public network; in private network just refers the connection within a company or a building. And rest of these networks comes under in the public access type. These security aspects involved in the organization environment, venture and institutions.

Main operation behind the network security is to protecting the network and protecting the operations in the system. Simple and easy way of protecting the data from unauthorized access is by assigning the unique identity and password. Each and every user has own id and password. Generally a security aspect initiates with the authenticating process by using the username and password; this idea arrived from the one detail authentication or may be called it as one factor authentication.

Normally three ways are followed to achieve the authentication process. One factor authentication is discussed previously and remaining things are as follows. Two factor authentications just show “something the user has” For example Mobile phone or Automatic teller machine card. And the final one three factor authentication has verifies that “something the user is” for example fingerprint, eye rise, retina.

After this authentication process, firewall allows only necessary files for the particular user otherwise denied the services for keeping the privacy. Sometimes they failed to remove Trojan virus and computer worms from the system. By overall network security possesses the guarantee level that all machines in the network are behave or working properly. This operation can involve preventing the unauthorized access; guaranteeing and securing the operations involved in the network. Main thing in computer network is in order to understand the background of securities in network. TCP/IP network protocol is mainly used for getting the connection information over the internet. Without security, it is impossible to do network communication across Local Area Network or through a Wi-Fi.

1.2 ADDRESS RESOLUTION PROTOCOL

The Address Resolution Protocol is used for discovering Link Layer Address or Physical Address or MAC address, associated within a given internet layer address. There are two types of messages in Ethernet ARP, they are ARP Request and ARP Reply.ARP request is request for the destination hardware address that is typically sent to all hosts. ARP reply in response, this gives the host the hardware address of the destination host. When any two computers need to communicate in the network, they need to identify each other uniquely. There are two types of addresses that are used to uniquely identify a host.

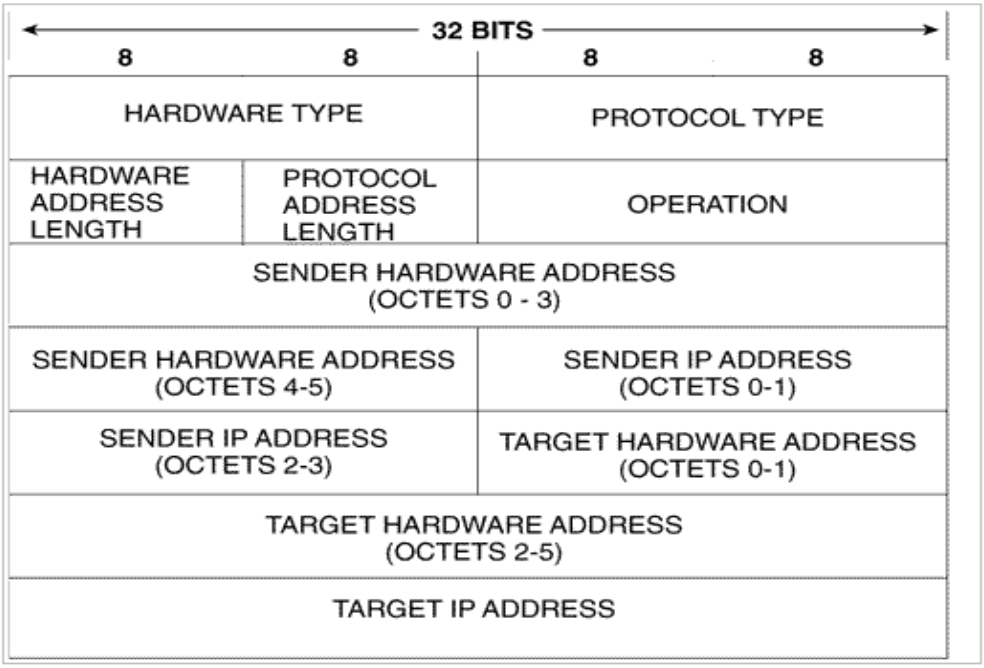


Fig 1.1 ARP packet format

MAC Address is also known as 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. When a host sends out an IP packet, it uses this source address and it receives all packets that match its own hardware address. This Ethernet address, typically a 48-bit address, is a link layer address and depends on the network interface card used. 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 Protocol (DHCP).

The individual networks that make internet are connected by routers. These routers need to know the IP addresses to send the packets in the right direction. Since routing rely on the IP addresses, but LAN use only MAC addresses. So, it would be quite not as convenient for each program to know both IP and MAC address. This is where the role of ARP comes into existence, for providing this so crucial <IP, MAC> association. So, administrators do not need to set these pairs but at the same time automation of these is complex issue.

The Address Resolution Protocol (ARP) is used by computers to map network addresses (IP) to physical addresses (MAC). When any host that wants to know the MAC (Media Access Control) address of another host in the LAN network, it broadcasts an ARP request in the network asking for the MAC address of a host with particular IP. The host with the given IP replies back in a unicast ARP replies along with its corresponding MAC address. The host that issued the request store that <IP, MAC> association in a local ARP cache so that it could use that pairing in the near future if required.

1.3 WORKING OF ARP

When an Ethernet frame is broadcasted from one machine on a LAN to another, the 48-bit MAC address is used to determine the interface for which the frame is destined. Address resolution refers to the process of dynamically finding a MAC address of a computer on a network. The protocol provides a dynamic mapping between the two different types of addresses that are IP address and MAC address which are used by data link layer. The process is dynamic since it happens automatically and is normally not a concern of either the application user or the system administrator. In a shared Ethernet where hosts use the TCP/IP suite for communication, IP packets need to be encapsulated in Ethernet frames before they can be transmitted on to the wire. There is a one-to-one mapping between the set of IP addresses and the set of Ethernet addresses. Before the packet can be encapsulated in an Ethernet frame, the host sending the packet needs the recipient's MAC address. Therefore, ARP is used to find the destination MAC address using the IP address.

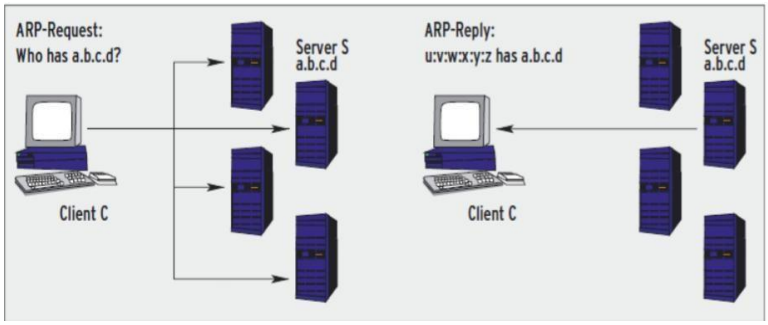


Fig 1.2 Working of ARP

In the figure, if client C needs to send a packet to server S, it needs to know the MAC address of S if both machines are on the same sub net. Even if S resides in a different network, C still needs the MAC address, in this case, the address of the next router that will forward the packet. The router takes care of everything else. To ascertain the MAC address, C broadcasts an ARP request to all the machines on the local network, asking "Who has the IP address a.b.c.d?" The

computer with the matching number replies and tells the client its MAC address. In order to minimize the number of ARP requests that are being broadcast, operating systems maintain a cache of ARP replies from different hosts. When a host receives any ARP reply, it will normally update its ARP cache with the new IP/MAC association entry. Note that the <IP, MAC> mapping received in the ARP reply should be used to update the ARP cache, only if that sender's IP address is already in the table . 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 sends the bogus replies to the communicating systems, thereby making the changes favourable to attacker, in the pairing of IP and MAC addresses. By doing this the information starts going through the attackers machine, without coming into notice of actual hosts. As a performance improvement, some operating systems (e.g., Linux and Windows) cache replies received from hosts whose IP addresses were not previously in the ARP table.

1.4 ARP SPOOFING

ARP has been proved to work satisfactorily under regular circumstances, but it was not designed in order to cope with malicious hosts. With ARP cache poisoning or ARP spoofing attacks, an intruder can easily impersonate another host and can get access to sensitive information. Furthermore, these attacks can be easily performed by using widely available and easy to handle tools specially designed for attacking purposes only. There are many security issues arise with the use of ARP in a LAN. It may create vulnerabilities and threat to the confidentiality of data, several schemes to mitigate, detect and prevent ARP attacks have been proposed. But each has its limitations. In this thesis we tried to work in a best possible way to identify a most reliable solution to the problem, analysing each of the schemes to look into their advantages and weaknesses.

ARP spoofing is mainly construction of forged ARP replies. When a forged ARP reply is sent, a target computer could be easily pursued to send frames meant for Host A to instead go to Host B. If done properly, Host A will have no idea that any such redirecting of data has taken place. The process of updating a target computer's ARP cache with a forged entry is referred to as "poisoning". The result of ARP cache poisoning is that the IP traffic intended for one host is diverted to a different host.

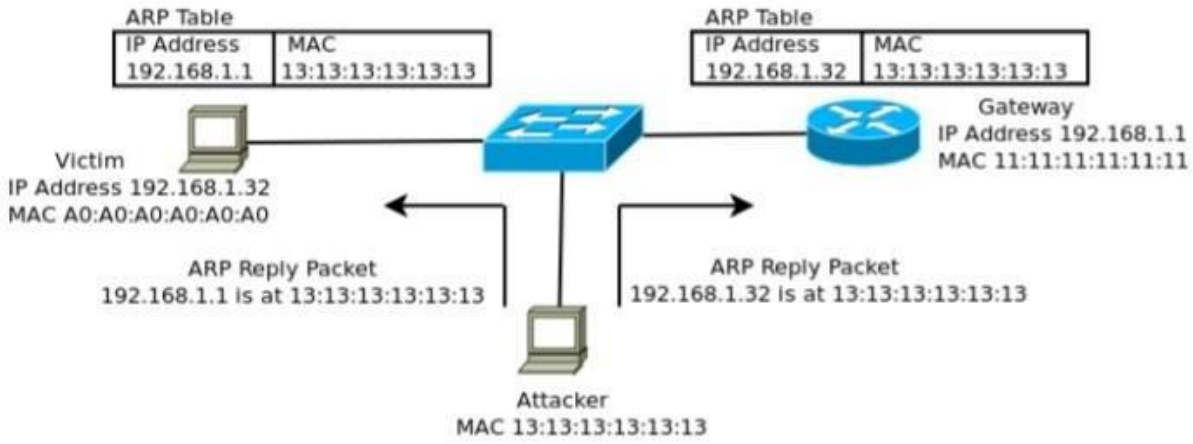


Fig 1.3 ARP spoofing, reply packet sent to victim and gateway

1.5 METHODS OF ARP SPOOFING

Unsolicited Response:

A response which is not associated with any ARP request will be honoured by such an ARP implementation attack. A rouge host only needs to send a response or reply ARP packet on the LAN with a spurious mapping in order to poison the ARP cache of victim host. Such a response can be broadcasted to poison cache of every host on the local network.

Request:

ARP caches the replies based on the requests it made. That means, if any host X sends out a broadcast of ARP request to know MAC address of host Y, host Z may store in its cache the mapping information regarding host X based on the request host X sends. An attacker only needs to pretend as if it is the one to send the legitimate request in order to poison the ARP cache of a victim host.

Response to a request:

Instead of sending an unsolicited response, or an unauthenticated request, a malicious user may choose to wait until a victim broadcast a request and

sends a response to that request. If legitimate host responds to the request, there is a race condition which the malicious host can win.

Request and response:

A malicious user may also send out both a untrue request and a response corresponding to that request. And this could be used to poison the victim's ARP cache in a case the victim remembers a request, either its own, or from another host and only caches a response to a request.

CHAPTER 2
2. LITERATURE SURVEY

Many efforts have been made and different methods have also been applied to prevent such attacks at ARP, but none has been able to give satisfactory results. Different tools and architectures have been proposed but each have their own feasibility issues.

Gibson Research Corporation discovered the problem of ARP cache poisoning. They found that problem in Ethernet is that all LANs are designed without any sort of authentication and in fact any computer can re-route all the of the LAN's traffic through itself but also edit and alter anything sent to or received from any other machine on the local network.

G.Gouda, Chin Tser [1], A secure address resolution protocol, Computer Networks suggested a protocol that can be used to solve the uncertified ARP table renewal from an ARP reply. This can overcome the problem of ARP spoofing, but it is practically impossible to modify the current protocol.

V.Ramachandran, S.Nandi [2], Detecting ARP spoofing: an active technique, Lecture Notes in Computer Science suggested effective search methods to detect and prevent ARP spoofing .However search methods are not able to solve the problem of ARP Spoofing in Local Area Network .The management of such passive methods requires the part of Network Administrators.

D.Pansa, T.Chomsiri [3], Architecture and protocols for secure LAN by using a software-level certificate and cancellation of ARP protocol, in:

Proceedings of International Conference on Convergence and Hybrid Information Technology (2008) suggested a method for installing a new DHCP server for use as a MAC-IP database server. They suggested a new DHCP for MAC address transmission between each host. However, DHCP also has been widely used and in practice, hard to fix and this method cannot be applied easily in a real life situation.

Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks proposed by Seung Yeob Nam, Sirojidin Djuraev, Minho Park [4](2013) suggested a Dynamic ARP Inspection (DAI) corresponds to an approach requiring the support of Ethernet switches that might prevent ARP poisoning. However this requires manual configuration by Network Managers.

Various Solutions for Address Resolution Protocol Spoofing Attacks intimates by S.Venkatramulu, Dr.C.V Guru Rao [5] (2013) differentiated various solutions for Address Resolution Protocols. It specifies number of tools to do ARP spoofing attack, this paper can be used a reference by researchers to decide how to safeguard the ARP protocol.

Detection and Prevention of ARP Cache Poisoning proposed by Neel Ravel, Payal Chaudhary (2015) [6] suggested that by detecting the ARP cache poisoning can minimize the ARP spoofing attacks.The detection malicious user who done malicious activity in network and perform MITM in network can be with the help of Snort .Here Snort is used as an IDS(Intrusion Detection System).

Detection and Prevention of ARP Poisoning in Dynamic IP configuration by Raviya Rupal D, Dhaval Satasiya, Hiresh Kumar, Archit Agarwal [7](2016) demonstrated a utility that provides a mechanism which is based on Internet Management Control Protocol which uses secondary cache for checking pair entry of IP-MAC respective of the system in the network.

Bruschi, D., Ornaghi, A., Rosti [8] suggested the use of 'S-ARP: a secure address resolution protocol' to prevent ARP spoofing .In SARP each host makes use of invite-accept protocol to periodically register its IP-MAC pairs are hashed by a message digest algorithm. However this approach, requires the medication of ARP protocol as the sender needs to sign each ARP message with its private key, and the receiver needs to verify the signature with sender's public key.The drawback of this approach is that it is ineffective against ARP DoS Attacks.

Gouda and Huang [9] proposed an architecture in which a secure server is connected to the Ethernet and the communications with the sever takes place by using invite-accept and request- reply protocols. All ARP communications occurs in between the host and the server and the replies are authenticated by using shared pair keys.

Kwon et al [10] suggested a similar approach that securely manages IP addresses in a distributed network. It makes use of an agent that retrieves IP-MAC pairs from a host and forwards them to manger to construct a reliable IP- MAC mapping.

Ortega et al [11] proposed a scheme that can be used to ARP small LANs. It consists of two elements, a server that ARP cache and a switch that blocks all ARP messages.

Lootah et al [12] implemented a secure IP-MAC address in which an ARP reply is generated with an attached signature when a request is issued. These tools are cheaper than the switches with port security but it has low response time when compared to the switches.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEMS

Secure ARP protocol has been used as a replacement for the ARP protocol is definitely a permanent solution for ARP spoofing but the biggest drawback is that we will have to make changes to the network of all the tasks. In SARP, each host uses an invite–accept protocol to periodically register its IP– MAC pairs in a secure server. IP–MAC pairs are hashed by a message digest algorithm. This approach, however, requires medication of the ARP protocol as the sender needs to sign each ARP message with its private key, and the receiver needs to verify the signature with the sender's public key. Secure ARP extends ARP with an integrity and authentication for ARP replies. In order to maintain compatibility with ARP, an additional header is inserted at the end of protocol standard messages to carry the authentication information. S-ARP has been implemented under the Linux Operating System. It consists of a kernel Patch and a user space daemon. The kernel patch removes the ARP packet list through the dev_remove_pack () function. The daemon can act as AKD or as a generic host depending upon the command line parameter passed to the protocol at launch. Hosts that run the S- ARP protocol will not accept the non-authenticated messages. Furthermore, the list of hosts running S-ARP must be given to every secured host that has to communicate with an unsecured one. It is intended to be used only during the transition phase to a full S-ARP enabled LAN.

3.1.1 Limitations

i) It is not very scalable as going for a stack upgrade across all available operating systems is something vendors and customers will not be happy about.

ii) As S-ARP makes use of Digital Signature Algorithm (DSA) it results in additional overhead of cryptographic calculations.

iii) This approach is ineffective against ARP DoS attacks.

iv) It often slows down the overall network throughput to an unacceptable level in practice.

v) The biggest drawback is that we will have to make changes to the network of all the tasks.

3.2 PROPOSED SYSTEM

Many approaches and methods were developed to detect and prevent against ARP spoofing attacks, but all of them have certain shortcomings. This idea of changing the MAC address of the default gateway into static, prevents the ARP spoofing by the attacker. To change the MAC of the default gateway into a static a software named “ARP DEFENDER” has been developed. It is built by using Python version 3.6 with PyQt framework. It also need PyQT 5 which is a module used to create Graphical User Interfaces and PypiWin32 lib for Python which is a extension module that provides access to Windows API functions. It not only sets Gateway MAC-address static but logs all hardware changes history, so that we can understand what is going on our network or in public Wi-Fi. It is developed for only on Windows 7/8/10 for now.

CHAPTER 4
SYSTEM SPECIFICATIONS

4.1 SOFTWARE REQUIREMENTS

Operating System : Windows 10

Language : Python

Environment : Ettercap, XARP

Ettercap Version : 0.7.4.1-Lazarus

Programming package : Python 3.7

4.2 HARDWARE REQUIREMENTS

Processor : Intel Core i8

Hard Disk : 512 GB

RAM : 4 GB

4.3 ABOUT THE SOFTWARE

PYTHON 3.7

Python 3.7 is the latest version released in the Python language. Since Python is an interpreter language it consists of both Python (also called as Python Interactive Shell) which is used to execute a single command and get the result. Python Shell waits for the input command from the user, when user enter the command the Shell execute and display the result. It also consists Python Idle which is an IDE (Integrated Development Environment) for Python. Idle provides a fully-featured text to create Python scripts that includes

features such as syntax highlighting, auto completion and smart indent. It also has a debugger with stepping and breakpoints features.

PyQT5

PyQt5 is a module that can be used to create Graphical User Interfaces.PyQt5 is not backwards compatible with PyQT4.It is one of the most used modules for creating GUI because of its simplicity. Another advantage is that is used to create complex GUI apps in a short time. We can simply drag our widgets to build the form.

Virtual Box

The main purpose of using Virtual Box is to run multiple Operating systems simultaneously. OS Virtual Box enables the user to run more than one operating system at a time. This way we can run a software written for one OS on another, such as Windows software on Linux or a Mac without rebooting it. Virtual Box has a lot of support because it's open-source and free. It has a better drag and drop between host and Virtual Machine (VM).Yet another advantage is that it offers an unlimited number of snapshots.

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions- Python 2 and Python 3. Both are quite different.

Some of the advantages of using python:

i) Emphasis on code readability, shorter codes, ease of writing

ii) Programmers can express logical concepts in fewer lines of code in comparison to languages such as C++ or Java.

iii) Python supports multiple programming paradigms, like object-oriented, imperative and functional programming or procedural.

iv)There exist inbuilt functions for almost all of the frequently used concepts.

Features

- Interpreted
- Platform Independent
- Free and open source, Redistributable
- Embeddable
- Robust
- Rich Library support

Ettercap

Ettercap is one of the tool which is used for ARP spoofing which permits malicious attackers to perform various attacks such as Man in The Middle Attack (MiTM) and Denial of Service etc. It features sniffing or spoofing of live connections, content filtering on the fly. It supports active and passive dissection of many protocols. Unlike many of the programs that are command line only. Ettercap features a graphical interface that's beginner- friendly. But sometimes results may vary, Ettercap is a great tool for newbies to get hang of network attacks like ARP spoofing.

XARP

XARP is one of the tool which is used for detection of ARP attacks. XARP uses active and passive modules that detect malicious attackers in the system. It is important as the Computer firewalls and OS security does not provide protection against ARP based attacks. As soon as XARP detects the ARP based attacks it shows alert on the screen.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

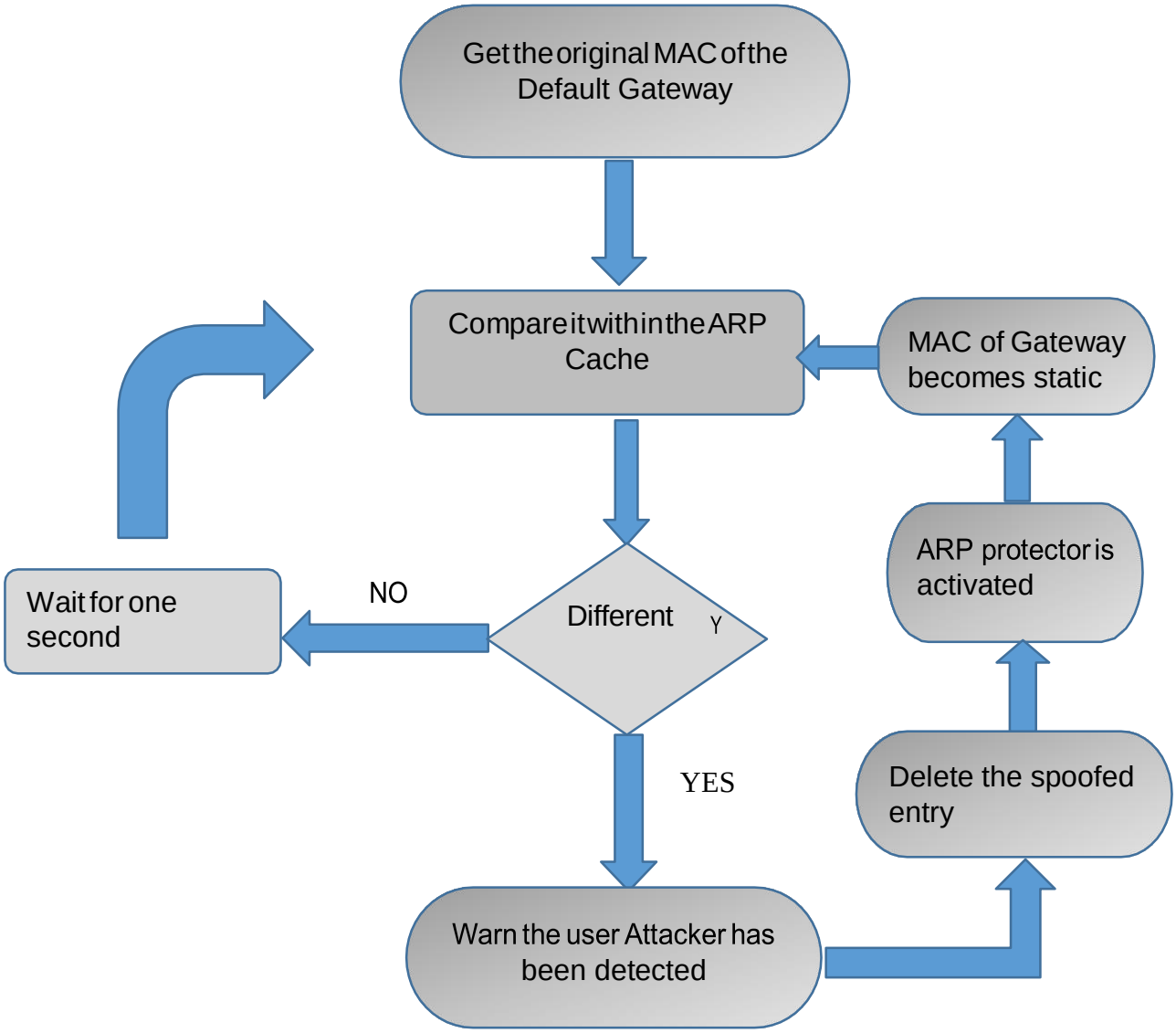


Fig 5.1 Architecture of the proposed system

5.2 MODULE DESCRIPTION

The proposed system consists of three modules:

1. Implementation of ARP spoofing
2. Detection of ARP spoofing
3. Prevention of ARP spoofing

5.2.1 Implementation of ARP Spoofing

In this section actual attack has been performed in the virtual environment to show how ARP cache poisoning happens in the network without the knowledge of the communicating hosts. Here it has been shown how different tools work to send bogus replies and made the cache to update the <IP, MAC> associations. The implementation of ARP spoofing can be performed by several softwares such as ARP-SK, ARPoison, Brian and Ettercap and so on. Ettercap is one of the most used tool for doing ARP based attacks. There are several steps to do ARP spoofing in Ettercap.

- Open Ettercap in your backtrack. Select the sniffing mode as Unified sniffing and give your network interface to be sniffed. Then scan for the host in your subnet.

- Corresponding hosts with their respective IP and MAC pairing has been shown. Now two target hosts are selected and added as Target 1 and Target 2. Victim with 192.168.43.1 IP is added to Target 1 and Victim with 192.168.43.244 to Target 2. For target 1 it is shown as below and for target 2, it can be done in similar way.

- Now, after the targets have been added, sniffing of the data can be started. The ‘start’ drop menu is chosen before selecting ‘Start sniffing’.

The MITM attack is performed using ARP poisoning attack. So, choose 'MITM' drop down menu and select 'ARP Poisoning'. In optional parameters to select, choose sniffing of remote connections. Thus Ettercap has completely spoofed two victims.

5.2.2 Detection of ARP Spoofing

There have been several tools such as XARP, ARPwatch, ARP Guard available free to detect the ARP Spoofing. XARP is one of the tool which is used for detection of ARP attacks. XARP uses active and passive modules that detect malicious attackers in the system. It is important as the Computer firewalls and OS security does not provide protection against ARP based attacks. As soon as XARP detects the ARP based attacks it shows alert on the screen. XARP inspect every ARP packet and report attacks against remote machines. Certain inspection modules can only work for the local machine (Static Preserve), but most of the modules does not need any local information. They monitor each ARP packet and it helps us in detecting the ARP attacks against other machines .XARP on a machine can see all network traffic from the whole subnet. XARP can monitor and inspect packets that it can see. XARP is unlikely to trigger alerts, it's a purely defensive tool with no offensive capabilities.

5.2.3 Prevention of ARP Spoofing

There has been several methods and approaches that has been developed to prevent ARP Spoofing but all of them have certain limitations. In this approach we try to make MAC of the default gateway into static, So that attacker could not penetrate the ARP cache of the system. When we often ARP Spoofing Detector, it has a Graphical User Interface from which user can inspect whether the default gateway has been spoofed or not. When a user finds out that the ARP spoofing has been occurred, the user prevents clicks the 'protect' button so that MAC of the Gateway becomes static. It begins with getting ARP table in a list of

records like (bool Static Flag) and it removes column ‘multicast’ and also skip the broadcast MAC-record. Then they get Internet Protocol Address and name of the Default Gateway. When it found that MAC of the Default Gateway has been spoofed, it removes the spoofed MAC and makes the MAC of the default gateway into static.

5.3 EXPERIMENTAL RESULTS

Thus above approach has protected the ARP spoofing attacks which prevents the malicious attacker could not penetrate into system and causing attacks. The MAC address of default gateway has been restored and it has been static, so that the attacker could not attack the system in future. It has been working successfully in Windows 7/8/10.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

From all the discussion we have seen that the matter of ARP spoofing is serious and it can lead to devastating results. With our mechanism that we have discussed, it has been shown that it is quite simple and easy to use. It could effectively prevent and detect the spoofing attacks being attempted in the Local Area Network. Their effect on the working of normal network has also been analysed and it does not have any performance degrading effects. Though, security do come at some cost, that we our self are injecting packets into the network. But it is better than being under attack of getting our crucial data invaded.

6.2 FUTURE WORKS

The protection of ARP spoofing has been limited upto safeguarding the Default Gateway. Still there is a lot to be researched in this area. We analyze the working of mechanism with different operating systems. More work could be done on improving false positive/negative ratios, when different hosts in the network are using personal firewalls in order to filter the data for both inbound and outbound. And to develop a solution with the use of integrated technology that could minimize the risk posed by the ARP spoofing.

APPENDICES

SAMPLE SOURCE CODE

Importing ARP Table

```
find_ip_addr = lambda x: re.findall(str_regex_ip, x)[0]
    find_mac_addr = lambda x: re.findall(str_regex_mac_addr, x)[0]
    find_flag_ip_multicast = lambda x: bool(re.search(str_regex_ip_multicast, x))
    find_flag_static = lambda x: bool(re.search(str_regex_static_flag, x))
    parse_output_line = lambda x: [find_flag_static(x), find_ip_addr(x),
find_mac_addr(x), find_flag_ip_multicast(x)]
    table = [parse_output_line(x) for x in res.splitlines()]
    filter_good = lambda x: x[1] and x[2] and not x[3]
    table = list(filter(filter_good, table))
```

Importing Default Gateway

```
res=subprocess.check_output('ipconfig', shell=True, universal_newlines=True)  #
alter: "netsh interface ipv4 show config"
str_regex_default_gw='Default
Gateway[^\d]+(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})|$'
find_default_gws = lambda x: re.findall(str_regex_default_gw, x)[0]
gateway_entries = map(find_default_gws, res.splitlines())
  gateway = next(filter(None, gateway_entries), '')
   return gateway
```

Finding IP Record

```
this_ip = lambda x: x[1] == ip
    record = filter(this_ip, table)
    default_record = [False, ip, '']
    return next(record, default_record)
find_gw_mac=lambda:find_ip_record(get_default_gateway_ip(),
get_arp_table())[2]
is_gw_static=lambdafind_ip_record(get_default_gateway_ip(), get_arp_table())[0]
```

Adding Static record

```
if not ip or not mac or not if_name:
        return
set_gw_static=lambda:add_static_record(get_default_gateway_ip(),
find_gw_mac(), get_default_gateway_interface_name())
def remove_static_record(ip):
    cmd = "Powershell Start-Process -WindowStyle hidden 'arp.exe' -ArgumentList
'-d %s' -Verb runAs" % ip
    subprocess.Popen(cmd)
set_gw_dynamic = lambda: remove_static_record(get_default_gateway_ip())
```

Clearing Static record

```
cmd = "Powershell Start-Process -WindowStyle hidden 'arp.exe' -ArgumentList '-d
%s' -Verb runAs" % ip subprocess.Popen(cmd)
set_gw_dynamic = lambda: remove_static_record(get_default_gateway_ip())
```

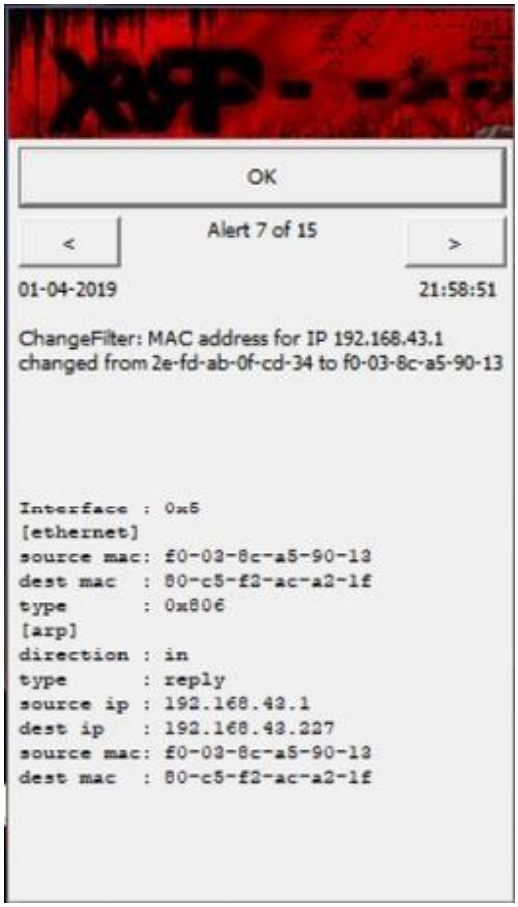
Updating ARP history

```
history = prev_history.copy()
for flag_static, ip, mac in table
if ip not in history.keys():
history[ip] = list([mac])
  if mac not in history[ip]
  history[ip].append(mac)
        return history
```

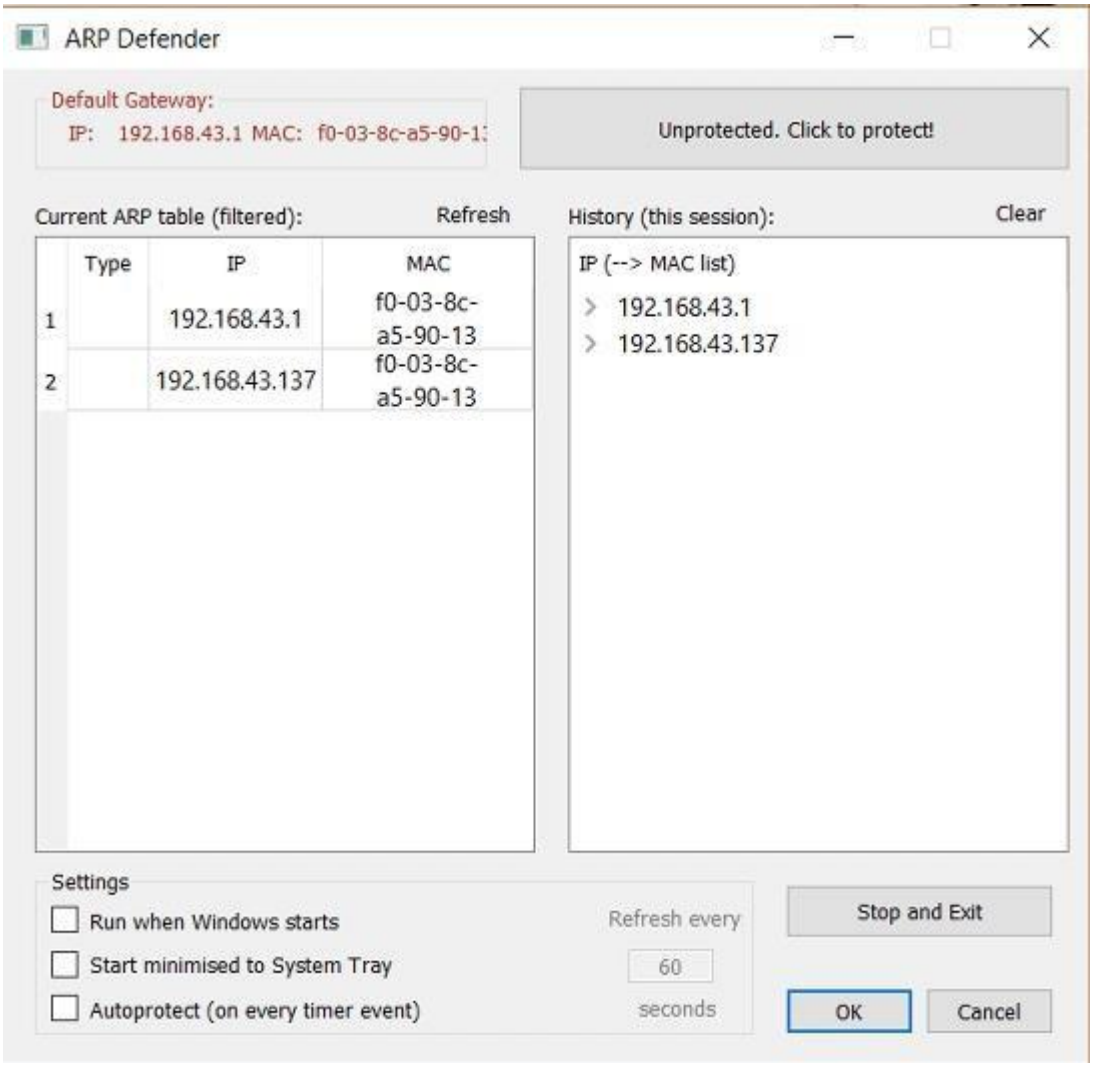
SAMPLE SCREENSHOTS

S1.Ettercap has been connected to the Network

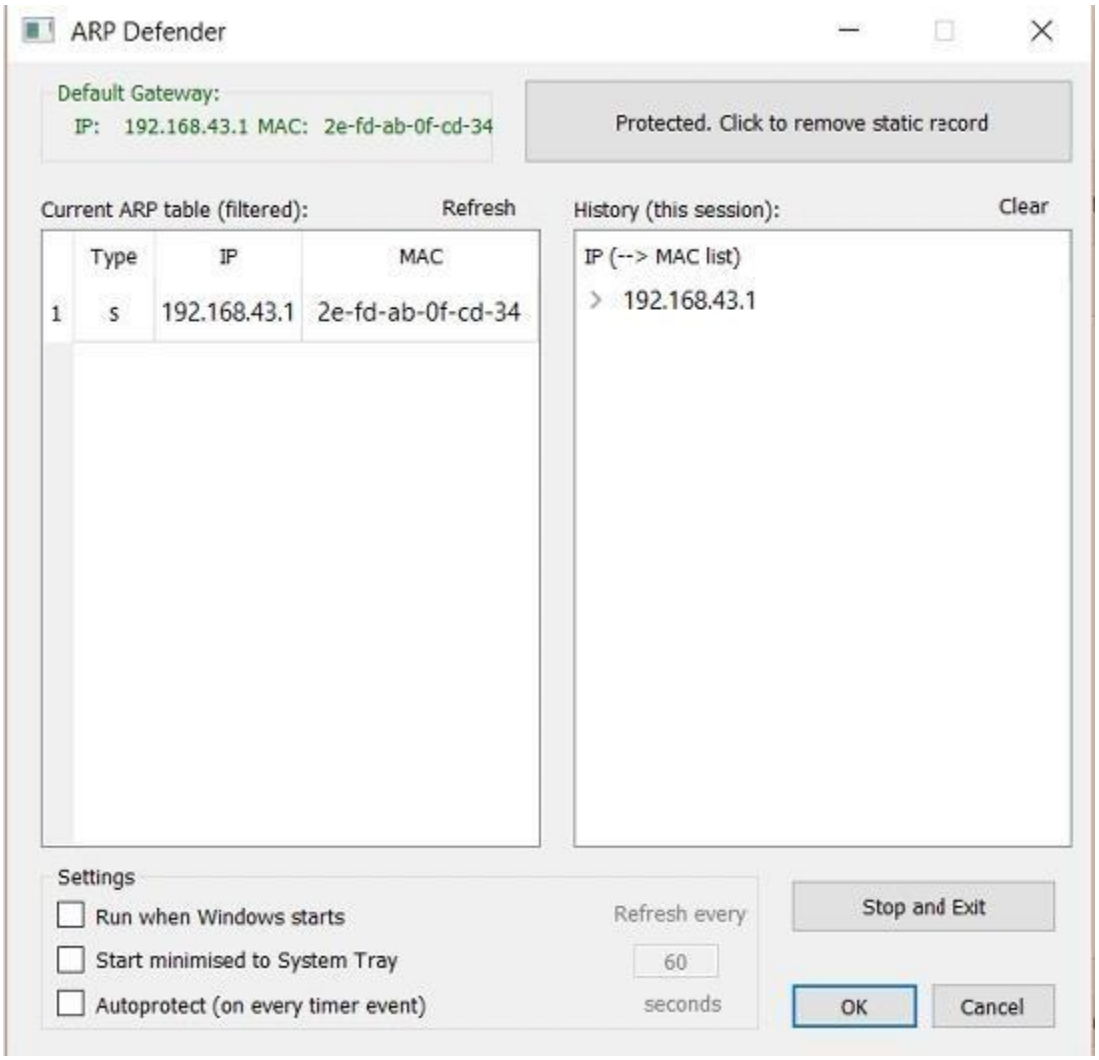
S2. Spoofing has been started in Ettercap



S3.Spoofing has been detected in XARP



S4. Spoofing has been detected and also found that attacker has an IP 192.168.43.137



S5. Spoofing has been prevented by making the Default gateway's MAC static.

REFERENCES

[1] G.Gouda, H.Chin-Tser, A secure address resolution protocol, Computer Networks 1 (41) (2003) 57–71.

[2] V.Ramachandran, S.Nandi, Detecting ARP spoofing: an active technique, Lecture Notes in Computer Science 3803 (2005) 239–250.

[3] D.Pansa, T.Chomsiri, Architecture and protocols for secure LAN by using a software- level certificate and cancellation of ARP protocol, in: Proceedings of International Conference on Convergence and Hybrid Information Technology,vol.2, 2008, pp.21–26.

[4] Seung Yeob Nam, Sirojidin Djuraev, Minho Park (2013) Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks.

[5] S.Venkatramulu, Dr.C.V Guru Rao (2013) Various Solutions for Address Resolution Protocol.

[6] Neel Ravel, Payal Chaudhary (2015) Detection and Prevention of ARP Cache Poisoning.

[7] Raviya Rupal D, Dhaval Satasiya, Hiresh Kumar, Archit Agarwal (2016) Detection and Prevention of ARP Poisoning in Dynamic IP Configuration.

[8] Bruschi, D., Ornaghi, A., Rosti, E.: 'S-ARP: a secure address resolution protocol'. Proc. 19th Annual Computer Security Applications Conf.(ACSAC2003), Las Vegas, NV, USA, December 2003, pp. 66–74.

[9] Gouda, M.G., Huang, C.T.: 'A secure address resolution protocol', Computer Network: Int. J. Computer. Telecommunication. Network, 2003, 41, pp. 57–71.

[10] Kwon, K., Ahn, S., Chung, J.W.: 'Network security management using ARP spoofing', Lect. Notes Computer Science, 2004, 3043, pp. 142–149.

[11] Ortega, A.P., Marcos, X.E., Chiang, L.D., Abad, and C.L: 'Preventing ARP cache poisoning attacks: a proof of concept using OpenWrt'. Proc.Network Operations and Management Symp. (APNOMS2009), September 2009.

[12] Lootah, W. Enck, W.Mcdanie, P: ‘TARP: ticket-based address resolution protocol’. Proc. 21st Annual Computer Security Applications Conf. on (ACSAC2005), Tucson, AZ, USA, December 2005, pp. 108–116.