



Security in Wired/Wireless Networks:
**Sniffing Attacks Prevention/Detection Techniques in
LAN networks & the effect on Biometric Technology**

A thesis submitted to The British University in Dubai, School of Informatics, in partial fulfillment of the requirements for the degree of M.Sc. in Information and Networking Security

**Faculty of Informatics
British University in Dubai**

Assessors
Board of Examiners

February, 2010

Author:

Moh'd Hussain Saleh Al-Hemairy,
School of Informatics, British University
in Dubai (BUiD)

Supervisors:

a. Internal Supervisor:

Dr. Saad Amin,
School of Informatics, British University
in Dubai (BUiD)

b. External Supervisor:

Dr. Zouheir Trabelsi,
Information Security Department, UAE
University (UAEU)

ABSTRACT

During the past era, Information Technology made a revolution in R&D. No doubt Internet becomes an essential backbone for all sciences and research nowadays. Accordingly security threats and data banks attacks turn out to be a phenomenon. Thus, granting protection to such crucial information becomes a high demand. While reviewing the latest studies in this area, there are strong signs that attacking information warehouse is the hot topic nowadays.

Moreover, preventing attacks to TCP/IP networks and what are the most efficient techniques to protect it, is the most targeted research area for security experts. For instance, what so called the Man-in-the-Middle attack [MiM] and Denial of Service [DoS] are just some ways of vulnerable attacks to TCP/IP networks, using some tools available free on the internet. They are sniffing the data traffic or causing service denial.

In our research, we evaluated the most famous security solutions and classifying them according to their efficiency against detecting or preventing the types of Address Resolution Protocol [ARP] Spoofing attacks. Based of the surprising experimental results done by a previous study in the security lab which proposed an optimal algorithm to enhance their ability against the two famous network attacks; we implemented the proposed algorithm by this study and simulate the experiment in order to test the algorithm performance. Moreover we studied the vulnerability of such attacks on two of the most famous security devices used for access-restricted sites; which is known as “Biometric devices”. The Biometric Technology includes Fingerprint readers and Eye readers [iRIS scanners] and we’ll show how they are performing against network attacks.

Keywords:

Sniffing Attacks, ARP cache poisoning, Man-in-the-Middle [MiM], Intrusion Prevention & Detection technique [IPS/IDS], Denial of Service [DoS], Fingerprint reader, Iris reader, Biometrics scanners

ACKNOWLEDGMENTS

From the formative stages of this thesis, to the final draft, I owe an immense debt of gratitude to my supervisor, Dr. Saad Amin at BUiD.

Dr. Zouheir Trabelsi has been the ideal thesis supervisor. His sage advice, insightful criticisms, and patient encouragement aided the writing of this thesis. His steadfast support of this project was greatly considered and deeply appreciated.

I express sincere appreciation to Dr. Khaled Shaalan at BUiD for his excellent courses he taught in master program and insight throughout the research.

I would also like to thank IT-College at UAEU, for offering access to their Labs and facilities to carry out our research and experiments, without their cooperation, this project would not have been possible.

To my colleague, Mr. Mohamed Ibrahim at Research-Affairs, for his inspiration and collaboration during my study

For her efforts and assistance, a special thank as well to my colleague, Ms. Huda Al-Hassani at College of Sciences, for her incentive support throughout my master study. Her knowledge and experience were of great help to me.

To each of the above, I extend my deepest appreciation.

DEDICATION

For my father who offered me tremendous support & encouragement to complete this thesis until this moment.

For my mother who motivated me to commence my postgraduate study and inspire me at the inauguration of my study.

For my wife who offered me care & back-up all the way through this thesis.

For a special person who fought to push me toward finishing this thesis.

TABLE OF CONTENTS

Description	Page
CHAPTER 1: INTRODUCTION	
1.1 Overview	5
1.1.1 What is an ARP	8
1.1.2 How it works: The ARP Process & RARP	8
1.1.3 Types of ARP/RARP protocol messages	9
1.1.4 ARP and RARP message formats	12
1.1.5 TCP Standard Ports/Services	13
CHAPTER 2: LITERATURE REVIEW	
2.1 Background	14
2.1.1 ARP Spoofing based on MiM and DoS attacks	14
2.1.2 How ARP caches are updated?	15
2.1.3 ARP Cache Poisoning (Spoofing) Attack	15
2.1.4 ARP Cache Spoofing (Example)	17
2.1.4.1 ARP Cache Spoofing (before ARP corruption)	17
2.1.4.2 ARP Cache Spoofing (after corruption)	18
2.1.5 Gratuitous ARP	19
2.1.6 MiM attack	19
2.1.7 Denial of Service [DoS]	21
2.2 Evaluation Of Common Intrusion Detection Systems & Intrusion Prevention Systems	22

2.2.1 ARP cache poisoning and MiM attacks	22
2.2.2 Detection of ARP cache poisoning attack	22
2.2.3 Packets sniffing and MiM attacks	22
2.2.4 Secure ARP Protocol: Prevention mechanisms	23
2.2.5 Protection mechanisms at the Application layer	24
2.2.6 Some mechanisms for external protection	24
CHAPTER 3: REAL-TIME DETECTION TECHNIQUE	
3.1 ARP Cache Poisoning	25
3.2 ARP spoofing based MiM and DoS attacks	26
3.3 Abnormal ARP Packets	27
3.4 Experiments: Comparing security solutions in the market	31
3.5 ARP Storm and Scan	32
3.6 Analyzing the experiment's result	33
CHAPTER 4: EXTENSION ALGORITHM FOR ARP SPOOFING DETECTION TECHNIQUE	
4.1 Optimal ARP Spoofing Detection Algorithm	34
4.2 Sample of the code implemented in C++ language	36
4.3 Testing and evaluation of the Algorithm	39
4.3.1 Results of the test experiments	40
4.3.2 Final Outcome of the Algorithms	40
4.4 Lab work	42
4.4.1 Packet Generation using FramelP Tool	42
4.4.2 Detecting Fake ARP packets using Snort IDS	45

CHAPTER 5: EFFECT OF COMMON NETWORK ATTACKS ON BIOMETRICS PERFORMANCE	
5.1 Biometric technologies	56
5.1.1 iRIS Scan	57
5.1.2 Fingerprint	57
5.1.3 Facial Recognition	57
5.1.4 What is Biometrics and how it works	58
5.1.5 Network Attacks against Biometrics	60
5.1.6 Land Attack	61
5.1.7 SYN Flood Attack	61
5.1.8 Teardrop Attack	62
5.1.9 UDP Flood Attack	63
5.1.10 ARP Cache Poisoning effect on Biometrics	64
5.1.11 Experiments: Common network attacks vs. Biometric Devices	65
5.1.12 Attacking Tools against Biometrics	66
5.1.13 DoS Attacks Results for Fingerprint Readers	67
5.1.14 DoS Attacks Results for iRIS Readers	68
5.1.15 ARP cache poisoning attack results for fingerprint and iRIS readers	70

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS	
Conclusion and Recommendations	71
Research Publishing	74
Appendixes	75
References	79

CHAPTER 1

INTRODUCTION

1.1 Overview

As we mentioned in the abstract section that this research is focusing on the internal attack within the local area network [LAN] which is forming the major and critical attacks which the network resources are exposed to according to recent studies conducted in the Information Security domain[1]. We will demonstrate two major attacks affecting the Internet users & the local network; The MiM attack[2] (Man-in-the-Middle Attack) and DoS (Denial-of-Service). There are many tools and softwares widely available and for free of cost which can carry out many attacks over the network and violate the privacy of users, such tools like Sniffers[3] monitors data traveling over a network, it either can be of authorized or unauthorized function. It was started initially as a Network Analyzer to help the Administrator to perform health check and maintain the network activities; however it is used today to redirect the traffic and access confidential files.

We will also show case how such network vulnerability can effect our lives in practice. The real live cases are the most famous physical authentication techniques used for “**Restricted & Permit-Access**” to staff, managers, visitors in sensitive areas such as; Military Bases, Embassies, Armies, and most likely the Airports..!

Nowadays, all the places we just mentioned above are using what is called; Biometrics scanners for organizing the authorized access to their premises and “Security Gates”.

The most used Biometrics devices used recently are; “Fingerprint readers” and “Iris reader

In this research we also investigate the effect of common network attacks on the performance of several fingerprint and iris readers. Experiments are conducted using Denial of Service attacks (DoSs), ARP cache poisoning attack and some other attacks. The Biometrics are vulnerable to some attacks and are secure toward other attacks due to significant reasons which we are going to discuss in the chapter 5.

It demonstrates also that biometric readers are easy targets for malicious network users since they lack security solutions, such as Firewalls, and Intrusion Detection/Prevention Systems...etc.

In our research we will also apply some attacking technique against: Fingerprint and iRIS scanners only and evaluate their performance against network threads.

Biometric reader manufacturers have been focusing on offering an easy to use and practical device with effective cost, low enrollment and recognition time, and low rate of false match and non-match. However, since these devices are as any network host with IP and MAC addresses, they may be the targets of malicious network users.

We will also present an overview of biometrics technologies. And we will discuss the network attacks that have been used in the experiments against several Biometrics Readers. Then we'll present and analyze the experiment's results. Finally is the conclusion.

The research plan we followed in our work presented here are as follows:

1. Address Resolution Protocol [ARP]
2. ARP Spoofing attack [Poisoning]
3. ARP Spoofing based MiM & DoS attacks
4. Optimal ARP Spoofing detection algorithm
5. Biometric Technologies
6. Network Attacks versus Biometric Security
7. Experiments; applying multiple attacks against Biometrics (e.g. DoS Attacks, Land Attack, SYN Flood Attack, Teardrop Attack, UDP Flood Attack, ARP Cache Poisoning Attack...etc)
8. Network Architecture
9. Attack Tools
10. Experiment results & analysis for Fingerprint
11. Experiment results & analysis for iRIS
12. ARP cache poisoning attack results for fingerprint and iris readers
13. Lab works &
14. Conclusion

1.1.1 What is an ARP:

The Address Resolution Protocol (ARP) [4] is used by computers to map network addresses (IP) to physical addresses or what is usually refer to: Media Access Control addresses (MAC).

It translates IP addresses to Ethernet MAC addresses and classified as a **Networking protocol** used to find host's address given its IP address. Some network expert consider it as a **DataLink Layer protocol** because it only operates on the local area network or point-to-point link that a host is connected to[5]. The Address Resolution Protocol (ARP) is documented in RFC 826[1] and later it was adopted by other media, such as FDDI[6]. For more details about Internet Protocols Suits; see appendix [1]

1.1.2 How it works: The ARP Process & RARP

As we stated formerly from an architecture perspective, ARP is a layer 3 function (Network), however in a programming perspective ARP is considered as layer 2 (Datalink) because it calls the LAN data like layer code. RARP is stand for; Reverse Address Resolution Protocol, and it is a network protocol used to resolve a MAC address to the corresponding network layer address, i.e. RARP is used to map a MAC address to an IP address exactly the reverse function of the ARP request/reply. The figure (1.1) below shows the location of ARP in the TCP/IP Architectural Model[7]:

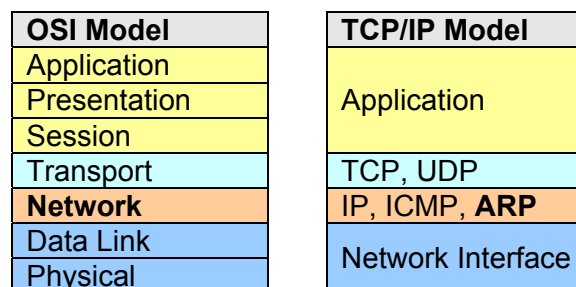


Fig.1.1: TCP/IP Architectural Model

1.1.3 Types of ARP/RARP protocol messages:

There are four types of ARP messages that are sent by an ARP protocol:

- a. ARP request
- b. ARP reply
- c. RARP request
- d. RARP reply

As we just said in the definition, ARP is used to map network address (IP) to physical address (MAC) and when a host need to communicate with another host it needs to know its MAC address. Here comes ARP protocol and works by broadcasting a packet (ARP-Request) for any hosts connected over the Ethernet network. The ARP packet contains the IP address of the sender and the IP address of the target it is interested in communicating with. See Figure (1.2) and Figure (1.3):

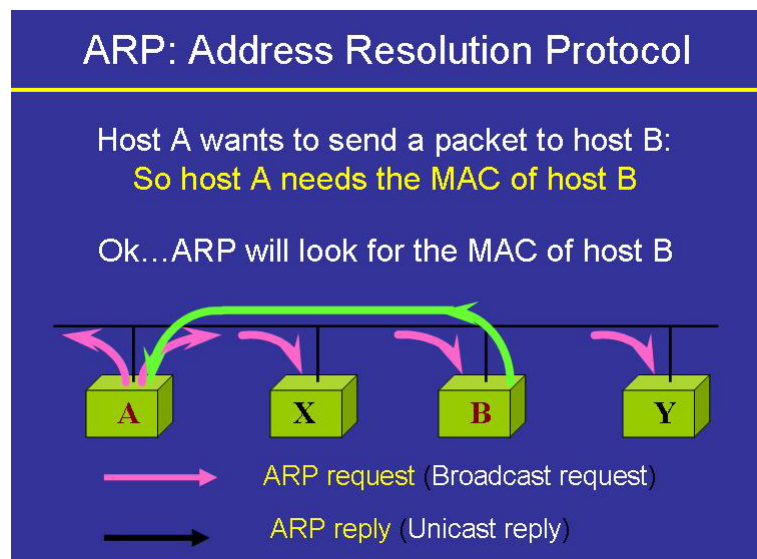


Fig.1.2: ARP: Address Resolution Protocol[8]

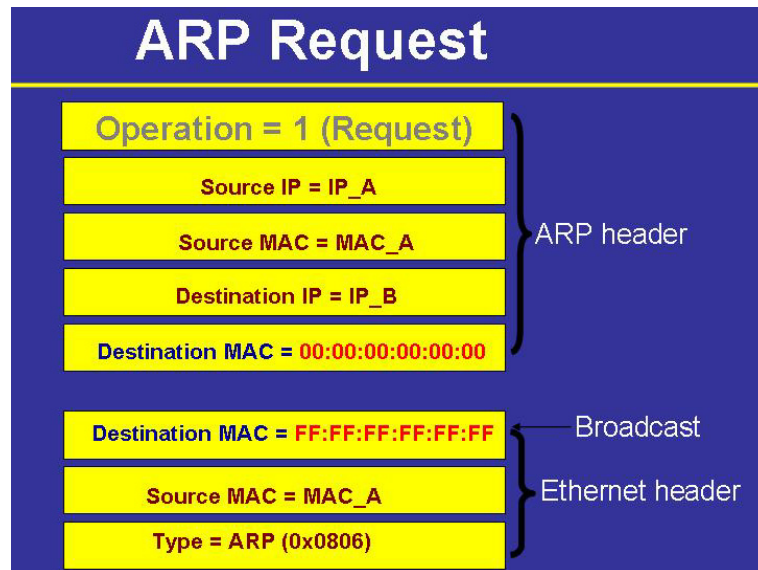


Fig.1.3: ARP request[8]

However, the target host, identifying that the IP address in the ARP request packet is belong to itself, so it returns an answer back in a unicast reply (ARP-Reply) and the host which initiated the ARP request catches the [IP,MAC] pair and keeps it in ARP cache memory. Keeping the host reply in cache will minimize the ARP traffic in the LAN. See Figure (1.4):

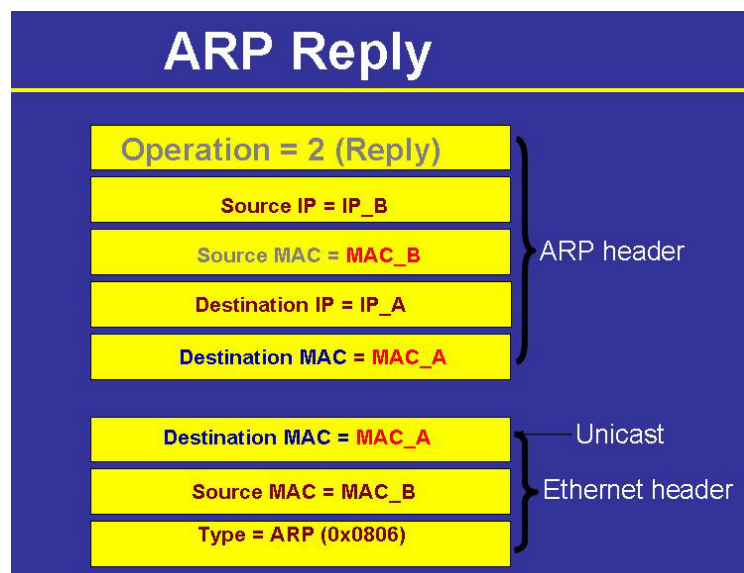


Fig. 1.4: ARP reply[8]

So simply when the ARP request is broadcasted to all PC's on the network it asks the following question:

- Is x.x.x.x is your IP address?, if Yes send back your MAC address.

Then every PC checks if its IP address is matching the one in ARP request and sends ARP reply with its MAC address.

But the repeated ARP requests especially when it is broadcasted every time a MAC address is required; creates a high traffic in the network, and hence the Operating Systems keep copy of the ARP replies in the computer's cache memory and update it frequently with any new *<IP, MAC>* pair, this will help in reducing the ARP requests number[9]. By the way ARP spoofing technique which we are going to talk about in the next chapter is occurring when forged ARP replies *<IP destination, MAC attacker>* is created and sent to the source computer who initiated the ARP request formerly and updated its ARP cache with fake information. We will know afterward this kind of exploitation is called "poisoning the ARP cache".

The Reverse Address Resolution Protocol [RARP] is broadcasting a RARP request packet with the target MAC address which will be received by all hosts in the Ethernet network. Host which its MAC address is matching the one in the RARP request will reply with its IP address in the RARP reply packet and sends it to the host which initiated the RARP request.

Afterward the IP address which consists of 32 bit will be converted to 48 bit Ethernet address, by the suitable encapsulation mechanism. This is the common practice for the Address Resolution Protocol (ARP), which is documented in RFC 826 [51]. ARP can be defined as an exchange process between the network layers on the Ethernet networks. This process will map the IP address to the "Link Layer" address whenever required. The Data Link Layer address is considered to be a hardware address (although it is challengeable) on Ethernet cards; where an IP address is considered as a logical address which can be assigned to any device connected to an Ethernet network. A Datalink layer addresses are known by some other names, so that;

- Ethernet Address
- Media Access Control (MAC) Address, and
- Hardware Addresses

However, the most accurate expression which was adopted by kernel side is "Link-Layer-Address" since it can be changed by some tools[50].

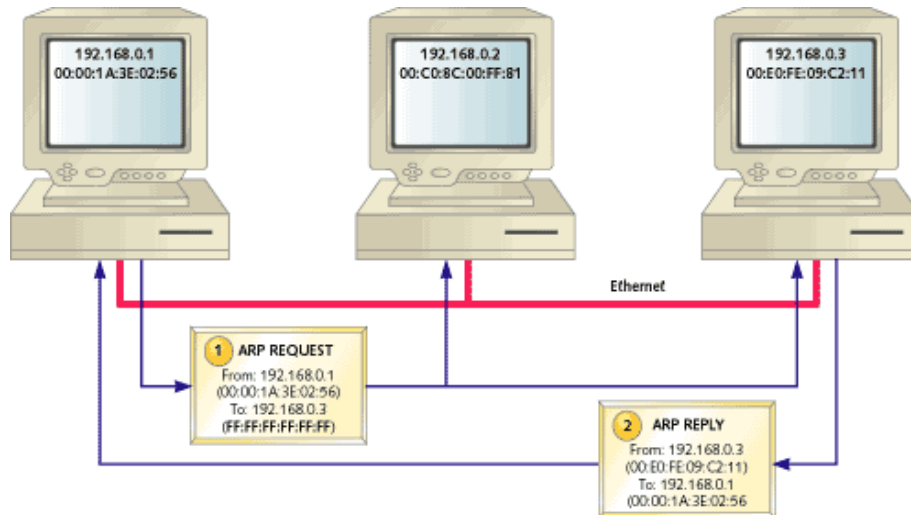


Fig. 1.5: ARP request and reply messages

1.1.4 ARP and RARP message formats:

The ARP packet consists of Ethernet Header and Data packet; the Ethernet header is divided to:

- 6 bytes for the destination address
- 6 bytes for source address
- 2 bytes for the frame type in hexadecimal (e.g. 0806 for ARP & 8035 for RARP)

Where, the data packet structure of ARP packet is encapsulated and the information that every part holds are demonstrated in the following table[10]:

Table 1.1: ARP and RARP packet structure

+	Bits 0 - 7	Bits 8 - 15	Bits 16 - 31
0	Hardware type (HTYPE)		Protocol type (PTYPE)
32	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
64	Source hardware address [MAC] (SHA) (first 32 bits)		
96	Source hardware address (last 16 bits)		Source protocol address (first 16 bits)
128	Sender protocol address (last 16 bits)		Destination hardware address (first 16 bits)
160	Destination hardware address (THA) (last 32 bits)		
192	Destination protocol address (TPA)		

- Hardware address type (2 bytes). 1=Ethernet
- Protocol address type (2 bytes). 0800H (hexadecimal) = IP address
- Operation type; 1 = ARP request, 2=ARP reply, 3=RARP request, 4=RARP reply
- etc....

1.1.5 TCP Standard Ports/Services

The table below is showing, a list of services and ports used by TCP protocol:

Table 1.2: TCP Ports and Services

Port #	Keywords	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	TelNet [Telecommunication network]
25	SMTP	Simple Mail Transfer
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80	HTTP	WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call

CHAPTER 2

LITERATURE REVIEW

2.1 Background

2.1.1 ARP Spoofing based on MiM and DoS attacks

ARP spoofing is also called; *ARP poison routing* (ARP) or *ARP cache poisoning* or *ARP Cache Corrupting*. It is a method of attacking an Ethernet local area network by updating the target ARP cache with a forged ARP request and reply packets[9]. This will try to change the target MAC address by another one which the attacker has a control on it. Updating ARP cache with a fake entry value is so called "ARP Poisoning".

What is sniffer? or (*The Network Analyzer*); it is a software or a hardware which log the traffic over a network and captures the data packets, then decodes the packets and analyzes the content. Kindly notice in our research that the following terms; ***Spoofing, Poisoning and Cache Corrupting*** are referring to the same term .

Furthermore, since ARP is considered as a trusted protocol within the network and is not designed to deal with malicious activities in the network, so attackers found unusual ways to illegitimately penetrate into the network; causing harmful costs.

These harms or costs can be much worse when the attacker tries to impersonate another user, performs Man-in-the-Middle attacks (MiM), or even causes Denial of Service (DoS) on a Server or even the whole Network[11].



P.S. Spoof means: hoax or imitation. Thanks to the British comedian Arthur Roberts (1852-1933), who introduced the word "***spoof***" to the world in the 19th century. He invented a game and called it Spoof, it incorporates tricks & nonsense[12].

Why it is so difficult to detect sniffers?

- The attack is essentially performed in the passive mode, which means it is hidden and working in the backend so the standard user will not recognize

such attacks. Besides it is not easily for user to detect the sniffing since this kind of attacks is generating usual traffic over the network.

- The other point is the fact that sniffers can be normally linked to an active intrusion attacks. While talking about the requirement and resources; sniffing is only requiring a standard machine connected over the network with normal hardware configurations and there is no need to special requirements or high performance.
- Threat is always seen as external and many researches shows that most of the attacks are from the internal resources; according to the recent Global security surveys in 2009[13], another study [14] shows that internal threat is incredible increased to more than 80% of the security breaches, where external attacks showed about 15% with internal help and 5% just from pure outsiders.

2.1.2 How ARP caches are updated?

Let us recall how the communication happens on an Ethernet LAN. As we early stated that all communications in layer 2 is based on the MAC address, so for any PC wants to talk to a target on the network is has to address it to the target's MAC address.

If a source computer tries to communicate with another computer in TCP/IP based network it has to translate the target's IP into the corresponding physical address (MAC) and here where we use an ARP protocol. The translation happens by request/reply ARP broadcast processes. When the ARP requester receives the reply, it catches the $\langle IP, MAC \rangle$ pair and keep it in it's ARP cache memory so it won't ask for it over again[15].

2.1.3 ARP Cache Poisoning (Spoofing) Attack

It is the process of corrupting an ARP cache with fake IP/MAC entries. It also used to perform some other attacks, for instance:

- Man-in-the-Middle (**MiM**) attack, also known as (**MITM**)
- Denial of Service (**DoS**) attack (refer to section 3.2)

As we discussed earlier if an entry is exist in the ARP cache, then it can be updated or corrupted using ARP reply or ARP request.

But what about if the entry; is NOT exist in the ARP cache?[16] The answer is: ARP request packets always work to corrupt any Operating System ARP cache whether the entry exists or not in the ARP cache. On the other hand, for hackers, ARP requests allow them to corrupt always the target ARP caches!

A recent study[16] showed by experiment the impact of the ARP request update on different Operating Systems. An experiment revealed which OS with dynamic entries in the ARP cache was vulnerable to the ARP cache poisoning attack[22].

Figure 2.1 [17], an evaluation for the impact of the ARP request update on different Operating Systems, e.g. Windows XP Professional, Windows 2000, 2003 Server, Linux 2.x, and Solaris 5.9:

Table 2.1: ARP request impact on various OS

	Windows XP		Windows 2000		Windows 2003 Server		Linux 2.4		Linux 2.6		Free BSD 4.11		SunOS Solaris5.9	
	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Entry exist in ARP cache?														
ARP request	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ARP reply	√	X	√	√	√	X	√	X	√	X	√	√	√	√
√ = ARP request or reply message is accepted by the system & allows the update or creation of MAC / IP entry X = ARP request or reply message is rejected by the system & does NOT allow update & creation MAC/IP entry														

And according to the study[22]; we can conduct the following:

1. Once the APR caches do not have an entry; the tested operation systems will not create a new entry by a reply message. This is with an exception case for Win 2K, Free BSD 4.11 & Sun Solaris 5.x.
2. Once the APR caches do not have an entry; the tested operation systems will create a new entry by a request message.
3. The operating systems will allow the update of the ARP reply if an entry is existed in the ARP cache.

The operating systems will find hardness to detect the ARP cache poisoning attacks, if the ARP Spoofing attack is using a reply ARP message. On the other hand it becomes much easier for the operating systems to detect an ARP request

messages. This means that many operating systems are exposed to this kind of attacks (cache poisoning)

2.1.4 ARP Cache Poisoning (examples)

The ARP Spoofing process is mainly to corrupt the ARP cache of any host over the network with fake IP/MAC pair in order to perform some serious attacks such as Man-in-the-Middle attack (MiM) or Denial-of-Service (DoS) [52]. In the following demonstration we will show the two different steps before and after the ARP cache poisoning is taking place, in the figure (2.1) and figure (2.2).

2.1.4.1 ARP Cache Spoofing (before ARP corruption)

In figure (2.1) it's clear that the ARP cache table is legitimate for all hosts connected to the network via a switch, where we can see that every IP is associated with a correspondence MAC for that host [53]. For instance; in ARP cache table of the host "A" ; the IP of the host "B" is mapped with the MAC of the host "B". And the same case is applied on host "C".

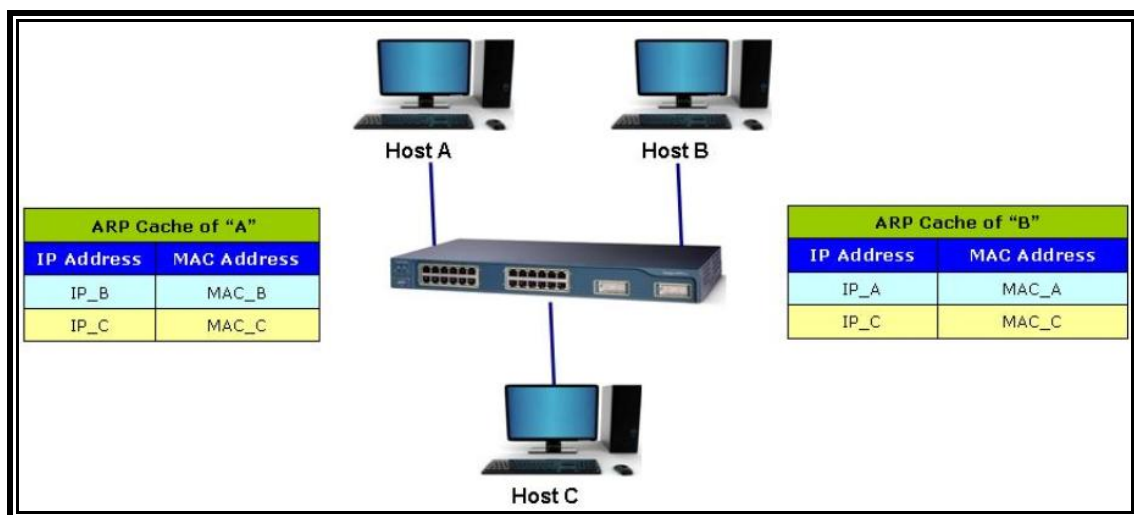


Fig. 2.1: ARP Cache Spoofing (before ARP corruption)

On the other hand, in ARP cache table of the host "B" for example; the IP-address of the host "A" is mapped with the MAC-address of the host "A". And the same case is applied on host "C".

- Let us see what changes may occur after the cache poisoning:

2.1.4.2 ARP Cache Spoofing (after corruption)

We can notice from the following figure that; the suspicious host is “C” in this scenario. The ARP cache tables for host A & host B are corrupted by the host C [54]. The ARP cache table for host “A” is becoming illegitimate now, where we can see that every IP-address is mapped to an invalid and not the corresponding MAC-address for that host. For instance; in ARP cache table of the host “A” the IP of host “B” is mapped with the MAC of the host “C” and so for host “B” [53].

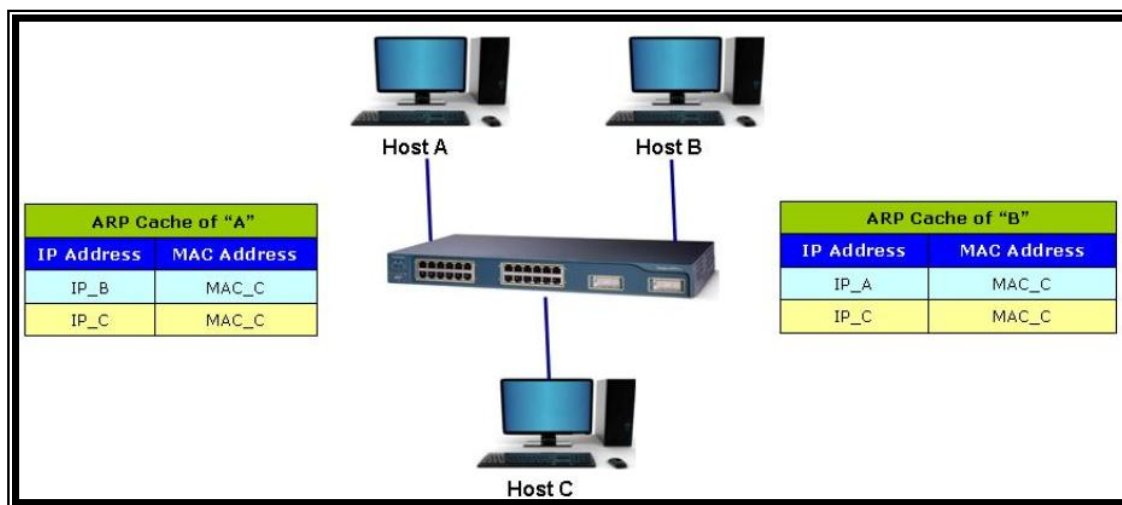


Fig. 2.2: ARP Cache Spoofing (after ARP corruption)

In this case whenever the host “A” want to communicate with host “B”, the TCP/IP traffic will be guided to pass by the malicious host “C” instead of “B”..!

So what..?

Hackers use the process of generating such abnormal ARP request packets to corrupt the ARP cache for certain hosts and perform different attacks over the network (e.g. MiM or DoS).

2.1.5 Gratuitous ARP:

This process is concerned about IP address duplication attack. Such a situation is due to the case when a host sends an ARP request to look for its MAC. This may occur when the host reboots, or once changing its Ethernet Number or the IP address[17].

Gratuitous ARP is doing the following tasks:

- i. Finding IP address conflicts in the Network by verifying if any host exist over the network and having a matching IP and displaying this message:
« duplicate IP address sent from Ethernet address: a:b:c:d:e:f » .
- ii. If a host changing its MAC or IP address by sending an ARP request, then it will force to update the ARP cache on the Network with the new MAC/IP address

P.S. ARP Gratuitous is mainly influence old Operation Systems, such as; Windows XP SP1 or older.

2.1.6 MiM attack:

The man-in-the-middle attack, (abbreviated as: MiM, or sometimes: MITM[18]) comes from the Packet-Sniffing[19]. MiM doesn't listen to all the packets that walk along the network as the Sniffer works, however it interfere with one or more hosts in the network and starts snooping between them. Such hosts been listened by a MiM are commonly called victims. A victim can be a normal host (e.g. PC or Notebook), gateway or even a router!

An attacker who is mainly spying between two or more victims; is establishing a autonomous connections between the victims and convey messages between them as if they are directly connected. And hence we call him: Man-in-the-Middle.

So far MiM is just listening to the traffic passing through two victims. Although this kind of outrage is illegitimate and can reach sensitive information like passwords, e-mail messages, encryption keys...etc. however it become worse and worse when he tries to go further than and inject false and fake packets and convey them between the deceived victims.

Figure (2.3) shows how MiM attack can be carried out in general:



Fig. 2.3: Man-in-the-Middle attack

According to [20] MiM attack is classified as an active attack, because the hacker manages the traffic in the network between the source and the destinations.

MiM is very famous approach used by hackers nowadays and uses the ARP protocol in order to attack the ARP-Cache tables and hence control the targets [21]. By poisoning the ARP tables for all hosts in the network for example; will instruct the hosts to reroute the traffic to the Attacker host instead of the Gateway, where he starts interfering between any two or more victims.

One more thing needs to be mentioned that the attacker has to forward all the interrupted packets to the original destination [16], so that the synchronized connection will remain and doesn't time out...!

Figure (2.4) is showing how the attack is taking place:

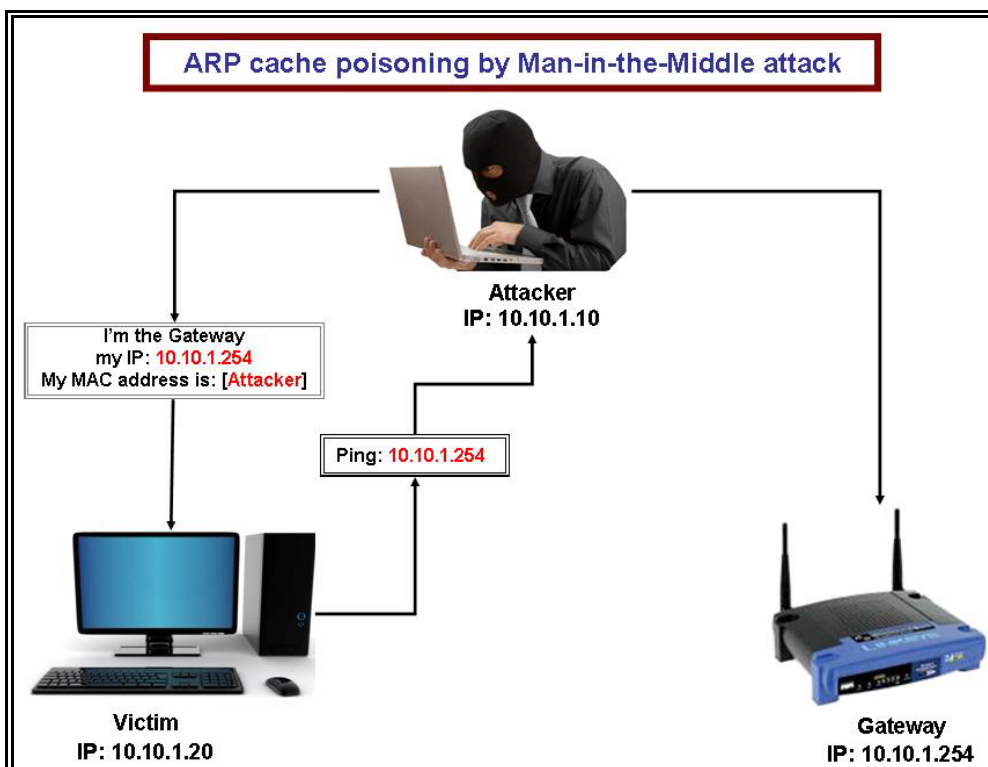


Fig. 2.4: illustrates how MiM-attack mechanism using ARP cache poisoning

In the above figure; ARP spoofing occurs when sending a fake and spoofed ARP reply to the target, i.e. if the Attacker has an IP: [10.10.1.10] and wants to sniff the traffic between the Victim who has an IP: [10.10.1.20] and the Gateway which has an IP: [10.10.1.254] it simply sends fake ARP replies to associate it's own MAC address with the Gateway IP [10.10.1.254]. The Victim then is trapped and starts sending all the packets intended to the Gateway to the Attacker address as in the above illustration.

2.1.7 Denial of Service [DoS]:

DoS attacks; occurring when any suspicious host over the network performs ARP cache poisoning and receives any packet designated to the original target to the suspicious host and cause a block in the connection between the host and the target which is being attacked. Kindly notice that more details regarding Denial of Service [DoS] will be discussed in section (3.2) in chapter No. 3.

2.2 Evaluation Of Common Intrusion Detection Systems And Intrusion Prevention Systems

2.2.1 ARP cache poisoning and MiM attacks:

The ARP cache spoofing attack and the Man-in-the-Middle attack are usually maintained and controlled by humans [22]. There are many solutions proposed in solving this type of security threat, depending on some techniques or protocols at different OSI model layers; such as; Application layer, Network layer and Data link layer [16].

2.2.2 Detection of ARP cache poisoning attack:

Arpwatch[23] and Snort[24] are examples of tools used to detect ARP Spoofing. They are working on the packet level. After monitoring the Ethernet activities and saves all logs of IP/MAC pairs [16].

Both ARPwatch & Snort are considered as sensors so that they need to connect to the monitor ports and look at all network activities [16].

2.2.3 Packets sniffing and MiM attacks:

On shared broadcast LAN networks, such as hubbed and wireless networks, packets sniffing can easily be achieved with minimal efforts. However, a switched LAN environment presents a different problem with few available techniques for sniffing. The first technique consists of connecting to an administrative port on the Switch and setting it to broadcast mode. The administrative port will now receive all traffic. A second technique is summarized by sending a large number of spoofed packets, which is usually an ARP packet (Address Resolution Protocol) to the Switch so it fails to open and sends all packets to all ports. However, a recent study[25] shows that only old switches models are vulnerable to this attack. Another technique, which is based on the MiM attack, is to tell target hosts on the LAN network to use an attacker's MAC address to get to any other host[53]. Man-in-the-Middle-attack [MiM] generates suspicious ARP flood over the network and the suspicious host receives all the traffic passes through the network, take a copy of every packet for himself and forwards the copied packets to it's original destination[16].

Today, security devices, such as IDS's (An intrusion detection system) [26] and IPS's (An Intrusion Prevention System)[27], are vital equipments for any security solution for conserving the IT components in any enterprise. Intrusion Detection System discovers the harmful attacks (for instance; DoS, IP Spoofing). However, it has limitations against numerous types of security attacks for example; Man-in-the-Middle attack. Prevention mechanisms, such as S-ARP[28] and O-ARP[29] lack efficient implementation on real systems and for a performance evaluation[22].

2.2.4 Secure ARP protocols: prevention mechanisms:

S--ARP[28] is a famous protocol which is using cryptography signed for ARP reply [16]. In brief, it's all about an entry will be updated when its signature is verified. However this process will slow down the network and have no significant protection on ARP Spoofing [16].

a. O-ARP technique:

O--ARP[29] is a new discovered protocol which is as close as to the secured S--ARP protocol. They are similar in the message format and key management[54]. The O--ARP is different than the S--APR in using the cryptography signature for ARP replies, it uses the cryptography when it is necessary only to minimize the network low performance. The study[29] showed that O--ARP can as fast as than S-ARP, and so it can used ARP Spoofing prevention attacks. Unfortunately the O--ARP is not yet implanted or tested on OS; so we can know how it performs against ARP Spoofing attacks.

b. Ticket-based Address Resolution Protocol

Ticket-based Address Resolution Protocol (T--ARP)[31]. It is considered as APR extension as a secured protocol.

The T--ARP distribute the IP and MAC addresses pairs by an ARP message in a secure way[54] using what is may considered "tickets". This method is less costly than other proposed solutions for ARP based issues, it require only single public key to validate a reply or request message only..!

There are clear disadvantages for this protocol; which is the weakness in defense against Denial of Service [DoS] attacks and the ticket flooding, besides it doesn't support Dynamic IP's in DHCP environment[54]

c. Cryptographic Technique

This method was shown in [32] study. It depends on digital signature and a password that generated once with chain of hash keys. Another protocol is the ARP-Sec; is uses a security key that is known only by packet source & destination [33]. This protocol can also used confidence and integrity, besides authenticating the traffic used by IP-protocol [34]

2.2.5 Protection mechanisms:

There were many mechanisms proposed to secure the application layer in the OSI network model. For instance; the papers [35], [36], [37],[38] introduced a variety of solutions to secure application layer, however with vain. They all have become vulnerable to an attack while they are trying to defense against the main attack type. The Password Protection Module (PPM); presented in [39] can protect against some types of Man-in-the-Middle attacks, however it is complicated to be implemented. A new theory presented in [40] for Man-in-the-Middle attacks but it has some limitations and lead to low network performance since it needs to create a secure tunnels with authentication which is costly in time and processing.

2.2.6 Some mechanisms for external protection:

An external method to secure the ARP protocol was presented in [41] it assumes to distribute all the IP addresses manually but this needs a dedicated System Administrators besides it load them with additional assignments. The security port in [42] was protecting against some types of hijacking the MAC addresses. However, it can be forged by the attacker.

CHAPTER 3

REAL-TIME DETECTION TECHNIQUE

There are numerous expensive Network Intrusion Protection Systems and Intrusion Detection Systems [IPS/IDS]. Such Hardware appliances or tools which are available in the market are expensive too and their manufacturers claim that their solution can handle the ARP Spoofing. In our research we evaluated the most renowned and common security solutions and experiment the competence for ARP Spoofing detection.

While analyzing the result of the experiments conducted in the Networking and Communication Lab at College of Information Technology in UAE-University for detecting ARP Spoofing [54], it was obvious that ARP Spoofing was not considered significantly by the security solutions we studied earlier, despite the fact ARP Spoofing is a serious danger upon networks while it is simple to execute. (Kindly refer to chapter 2: ARP Spoofing)

The experiments showed also that many Detection and Protection appliances and tools are not fully robust against ARP Spoofing attacks. In Experiment section underneath, table 4: shows the performance of the security Appliances against 10 different ARP Spoofing attacks. Thus, and as a solution for this “flaw” in these systems we extended the Algorithm that was proposed in the study [54] and implemented it in a simulation environment so that any security solution can equipped with in the future as a real-time detection technique in order to detect ARP Spoofing attacks.

In the following sections; we will introduce how the authors in[54] conducted a fake ARP Spoofing attacks (10 types) and performed an experiment on the Security Solutions which are dealing with Intrusions in the network. After that the experiment results will be analyzed. And then we will extend the algorithm as a real-time detection technique for ARP Spoofing

3.1 ARP Cache Poisoning

It is the process of mapping a forged IP host address to a valid host MAC address in the cache of ARP. The ARP poisoning can be done either by updating an existing

ARP entry or inserting new forged entry in the ARP cache for a target host. So in brief; there are two ways for performing ARP cache poisoning, the first one is adding a forged entry in the destination ARP cache, and the second one is updating an existing entry with a forged IP address or MAC address.

- **Adding a new forged entry** : an ARP request message with forged source IP and MAC addresses in the ARP header will be sent to the target host and when the host is receiving the ARP request message, it assume a connection establishes between the source and the host, so a new ARP entry is created in the ARP-cache[54]. And hence, the host ARP-cache will become corrupted by a forged IP and MAC addresses.
- **Update an ARP-cache with a forged entry**: although the MAC address & IP address of the target host are registered in the ARP-cache entries, it can be updated again by sending the forged IP and MAC again to the host target.

3.2 ARP spoofing based MiM and DoS attacks

Man-in-the-Middle attacks [MiM] and Denial of Service attacks [DoS] are mostly familiar in Local Area Networks, they are easy to launch as well.

When the malicious host re-forwarding the network traffic between the target hosts and is enabling IP Packet routing and starts sniffing the network; then MiM is taking place. The malicious host in this case is becoming similar to a router where it redirects all the traffic without any interruption as in Figure (3.1):

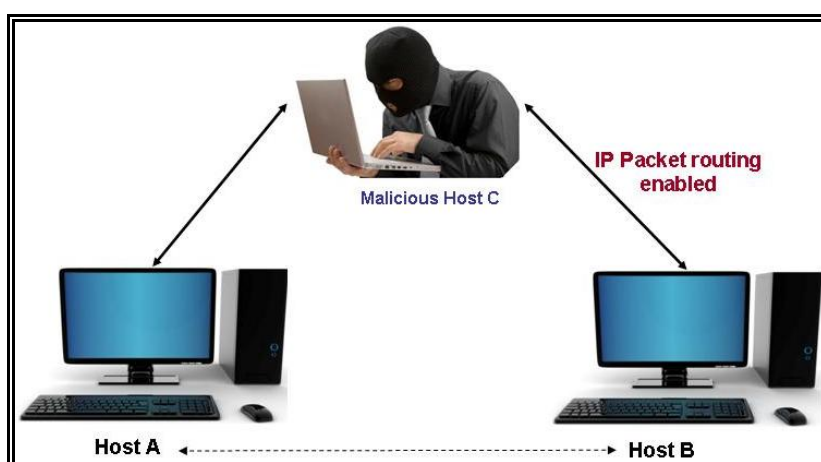


Fig. 3.1: A presentation of the MiM attack

In the ARP cache of the both targets was corrupted by the suspicious host, and if the IP-Packet routing was not enabled in the suspicious host, then there will be a Denial-of-Service case. So in the above figure; the IP-Packet routing is enabled in the malicious host “C”. And since the ARP cache is corrupted between host “A” and host “B” , then this case is called: “Man-in-the-Middle” attack.

The second case is the Denial of Service attack, in (figure 3.2). Host “A” and host “B” can’t communicate each others, since their ARP caches are poisoned by fake entries besides the IP-Packet routing is disabled in the suspicious host “C” and hence all the traffic between A and B will not be forwarded to the actual destination.

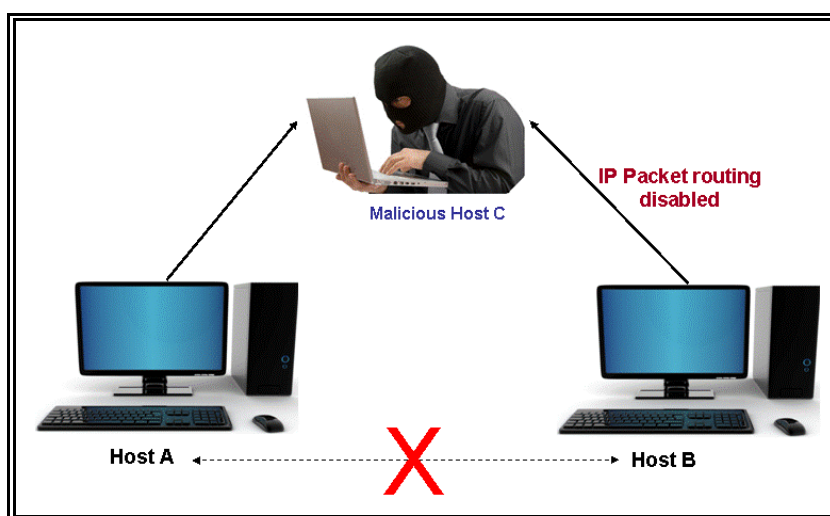


Fig. 3.2: A presentation of the *DoS* attack

3.3 Abnormal ARP Packets

To corrupt an ARP cache of any host over the network, an abnormal packet is used. There are two different types of these abnormal ARP packets; request packets and reply packets.

The research study in[54] identified the details of each type of abnormal packets for ARP request and ARP reply. There were four doable abnormal types of ARP request and six doable abnormal for ARP reply. Herewith the description and features of each type as shown in Figure 3.1 and Figure 3.2

Any ARP request encloses a mapping of IP-address and MAC-address. And these mapping is saved in a database for future use. So any break into these IP-MAC mappings can launch an alarm for the security device. The mapping can be done

manually or automatically. And these can be applied for the abnormal packets; [P1, P5, & P7]

For the abnormal packets; [P2, P6 & P8], there is a constraint for matching the Ethernet MAC of the destination with the ARP MAC of the destination in every ARP reply message. While the Ethernet source MAC address should match APR MAC address of the source.

An ARP request has to be sent in broadcasting mode for MAC address. However, if it is in a unicast mode then it is malicious request. An example for such case is [P3].

The tables (3.1) and (3.2), shows[54] clearly all the 10 types of possible ARP abnormal packets and what exactly is the problem in each case. It is obvious that only [P1 & P5] are able to cause ARP corruption to the cache with invalid IP-address and MAC-address entries. The other types of the abnormal packets can still causing Denial-of-Service attacks.

Table 3.1 : List of possible abnormal ARP request packets

Packet number	P#1	P#2	P#3 Unicast ARP request	P#4 Unexpected IP or MAC address in ARP request packets
ARP Header				
ARP Operation	1	1	1	1
Source IP	IP_A	IP_A		0.0.0.0 255.255.255.255 Multicast Not in the network subnet
Source MAC	MAC_X	MAC_A		00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast
Destination IP				0.0.0.0 255.255.255.255 Multicast Not in the network subnet
Destination MAC				
Ethernet Header				
Source MAC		MAC_X		00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast MAC
Destination MAC			Unicast	00-00-00-00-00-00 Unicast or Multicast
Does the packet corrupt the ARP cache?	Yes	No	No	No

IP_A: is the IP address of a host A; MAC_A: is the MAC address of a host A ;
 MAC_X: is a MAC address of a nonexistent host; Unexpected IP or MAC
 address in ARP request packets

Table 3.2 : List of possible abnormal ARP reply packets

Packet number	P#5	P#6	P#7	P#8	P#9 Broadcast ARP reply	P#10 Unexpected IP or MAC address
ARP Header						
Operation	2	2	2	2	2	2
Source IP	IP_A	IP_A				0.0.0.0 255.255.255.255 Multicast Not in the network subnet
Source MAC	MAC_X	MAC_A				00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast
Destination IP			IP_B	IP_B		0.0.0.0 255.255.255.255 Multicast Not in the network subnet
Destination MAC			MAC_X	MAC_B		00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast
Ethernet Header						
Source MAC		MAC_X				00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast
Destination MAC				MAC_X	ff-ff-ff-ff-ff-ff	00-00-00-00-00-00 ff-ff-ff-ff-ff-ff Multicast
Does the packet corrupt ARP cache?	Yes	No	No	No	No	No

IP_B: is the IP address of a host B; MAC_B: is the MAC address of a host B; Unexpected IP or MAC address in ARP reply packets.

3.4 Experiments: comparing security solutions in the market

In the study [54], they performed experiments on more than 10 most famous security solutions in the market, as in the following classification:

1. LAN switches
 - a. Cisco switch 3560 Series
 - b. Juniper Switches EX3200 Series
2. Software IDS/IPS
 - a. Snort IDS
 - b. XArp 2 tool
 - c. Sax2 NIDS
3. IDS/IPS hardware appliances
 - a. Cisco IPS 4255 Series
 - b. TopLayer Model 5000
 - c. IBM ISS Proventia Model GX4004C
 - d. SourceFire
 - e. TippingPoint 50
4. Unified Threat Management (UTM) devices
 - a. Juniper Netscreen 50

And the following table (3.3) demonstrates which solution could perform a successful APR inspection:

Table 3.3: Security solutions performing ARP inspection

Security solutions	Type	Performing ARP inspection (Yes or No)?	Detection or prevention solution?
Cisco Switch 3560 Series	Switch	Yes	Prevention
Juniper Switches EX3200 Series	Switch	Yes	Prevention
Snort IDS	IDS software tool	Yes	Detection
XArp 2 tool	IDS software tool	Yes	Detection
Sax2 NIDS	IDS software tool	Yes	Detection
Cisco IPS 4425 Series	IPS appliance	Yes	Detection
TopLayer Model 5000	IPS appliance	No	Detection
IBM ISS Proventia Model GX4004C	IPS appliance	No	Detection
SourceFire	IPS appliance	No	Detection
TippingPoint 50	IPS appliance	Yes	Detection
Juniper Netscreen 50	UTM	No	Detection

The security solutions which don't perform any ARP inspection were excluded. And Table (3.4) shows which solution performs total detection and which can only perform partial detection.

Table 3.4: Detection of abnormal ARP request and reply packets

	P#1	P#2	P#3	P#4	P#5	P#6	P#7	P#8	P#9	P#10
Cisco Switch 3560 Series	D	D	N/D	N/D	D	D	D	D	N/D	N/D
Juniper Switches EX3200	D	D	N/D	N/D	D	D	D	D	N/D	N/D
Snort IDS	D	D	D	N/D	D	D	D	D	N/D	N/D
XArp 2 tool	D	D	D	Partially	D	D	D	D	D	Partially
Sax2 NIDS	N/D	N/D	N/D	N/D	N/D	N/D	N/D	N/D	N/D	N/D
Cisco IPS Series 4255	D	N/D	N/D	Partially	D	N/D	D	N/D	N/D	Partially

D: for detection, N/D: for not detection, Partially: for partial detected

The study [54] showed that there were none perfect security solution against ARP Cache Poisoning. XARP2 tool looks perform better than others in regards to the number of detections against Abnormal ARP packets.

The study [54] showed that there were none perfect security solution against ARP Cache Poisoning. XARP2 tool looks perform better than others in regards to the number of detections against Abnormal ARP packets. Cisco switches 3560 [44] and Juniper switches EX3200 [45], seems to be promising against ARP Spoofing, however they are considered as costly solutions[46].

3.5 ARP Storm and Scan:

When an ARP entry is referenced in the cache memory and stays for few minutes (depending on the network configuration) it will be removed. This feature is misused by the malicious host to initiate what is called: ARP Storm. The attacker starts sending a continuous stream of fake ARP request packets to the target host. While ARP scan uses an ARP protocol to find out IP-address on the LAN. It can be a tool or malicious software such as; virus for instance. The table 3.6 demonstrates [54] which security solution can deal with ARP storm and ARP scan.

Table 3.6: Security solutions with ARP request storm and ARP scan detection mechanisms

	Detect ARP Request Storm?	Detect ARP Scan?
Cisco Switch 3560 Series	No	No
Juniper Switches EX3200 Series	No	No
Snort IDS	No	No
XArp 2 tool	No	No
Sax2 IDS	Yes	Yes
Cisco IPS 4425 Series	No	No

3.6 Analyzing the experiment's result;

The research done in[54] showed that the most famous and expensive security solutions have issues with detecting ARP spoofing. And hence, it proposed an algorithm to enhance any security device toward detecting all types of abnormal ARP spoofing attacks, such as: Man-in-the-Middle attack and Denial-of-Service attack.

Regarding cross-layer inspection technique, only few devices have the ability to perform it and therefore are protected against cache corruption.

According to the study[54], it claimed that the security tool XArp2 was the most sufficient one among the other solutions and just need some modifications in order to perform perfectly as it is shown in (Figure 3.5). Where XArp2 was connected to a SPAN port for mirroring and explore the network traffic. However, it can only detect ARP spoofing attacks not to prevent it..!!

To perfectly detect and prevent ARP attacks; a Cisco 3xxx Series[44] or Juniper EX32xx Series[45] advanced security switches are needed to be implanted even thought these are excellent solutions but they are expensive solutions.

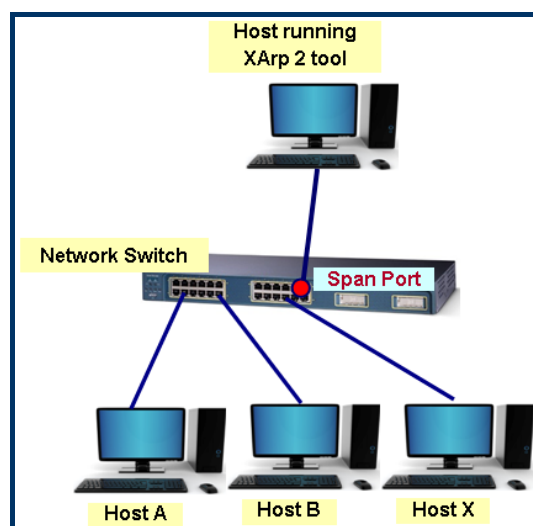


Fig. 3.5: an XArp_2 technique used in local area network[54]

CHAPTER 4

Extension on the algorithm for ARP Spoofing detection technique proposed in the study[54]

4.1 Optimal ARP Spoofing Detection Algorithm

The study claims that any security device intend to work against ARP Spoofing should have an advanced algorithm implemented in the box of any switch or security device in order to be considered as sufficient as for such malicious attacks. The minimum requirements for such algorithm are as follows;

- It should performs;

1. Cross-Layer ARP inspection for ARP header & Ethernet layer
2. Inspection for ARP statefull.

- And it can detect the following hoaxes;

1. None IP address and MAC address matching.
2. ARP Storm and,
3. ARP Scan. (refer to section: 3.5)
4. Making the mapping table between IP addresses and MAC addresses, either automatically for DHCP settings or manually by the network administrator in None-DHCP settings.

The following is the extended algorithm that was implemented in order to detect ARP Spoofing, presented in figure (3.6):

```

Abnormal Arp Packet Detection (Ethernet_header, ARP_header,
IP_MAC_Mapping_Table)

{ /* ARP request packet */

if (ARP_Operation = "request"):
{ Unicast_ARP_request (Ethernet_MAC_Destination); /* for detecting packet P#3 */

Unexpected_IP_MAC_Addresses_in_ARP_Request (Ethernet_MAC_Source,
Ethernet_MAC_Destination, ARP_IP_Source, ARP_MAC_Source, ARP_IP_Destination,
ARP_MAC_Destination);
/* for detecting packet P#4 */

Cross_Layer_Inspection_ARP_Request (Ethernet_MAC_Source, ARP_MAC_Source)
/* for detecting packet P#2 */

IP_to_MAC_Address_Mappings_ARP_Request (ARP_IP_Source, ARP_MAC_Source,
IP_MAC_Mapping_Table )
/* for detecting packet P#1 */}

/* ARP reply packet */

if (ARP_Operation = "reply"):
{ Broadcast_ARP_reply (Ethernet_MAC_Destination)
/* for detecting packet P#9 */

Unexpected_IP_MAC_Addresses_in_ARP_Reply (Ethernet_MAC_Source,
Ethernet_MAC_Destination, ARP_IP_Source, ARP_MAC_Source, ARP_IP_Destination,
ARP_MAC_Destination); /* for detecting packet P#10 */
Cross_Layer_Inspection_ARP_Reply (Ethernet_MAC_Source, ARP_MAC_Source,
Ethernet_MAC_Destination, ARP_MAC_Destination )
/* for detecting packets P#6 and P#8 */

IP_to_MAC_Address_Mappings_ARP_Reply (ARP_IP_Source, ARP_MAC_Source,
ARP_IP_Destination, ARP_MAC_Destination, IP_MAC_Mapping_Table )
/* for detecting packets P#5 and P#7 */
}}

```

Fig. 3.6: Optimal ARP Spoofing Detection Algorithm

- imagine the *IP_MAC_Mapping_Table* to look like the following table; where IP address in the row (i) and column (j) is represented by $[IP_{i,j}]$ and is mapped with the corresponding MAC in the row (i) and column (j), which is also represented by $[MAC_{i,j}]$. the figure 3.7 is showing how this mapping matches:

i/j	j	
	IP Address	MAC Address
i	IP _{1,1}	MAC _{1,2}
	IP _{2,1}	MAC _{2,2}
	IP _{i,j}	MAC _{i,j}

Fig. 3.7: Mapping table between IP-address and MAC-address

4.2 Sample of the code implemented in C++ language:

In the following figure (3.8) we will show part of the code implemented in C++ based on the algorithms that was presented in figure (3.6). It has been written in C++ language in order to test it in the LAB for detecting ARP Spoofed & Abnormal packets and to ensure that all known types of ARP Spoofing attacks are addressed and detected by the security solution. Section 3.9.3 shows the testing experiment:

```

Abnormal_Arp_Packet_Detection (Ethernet_header, ARP_header,
IP_MAC_Mapping_Table)

{ /* ARP request packet */

if (ARP_Operation = "request");
{
IP_to_MAC_Address_Mappings_ARP_Request (ARP_IP_Source, ARP_MAC_Source,
IP_MAC_Mapping_Table )
{ for i > 0 , j > 0 , i ++ , j ++;
    if (ARP_IP_Source = IP_MAC_Mapping_Table [i,j]) ;
    AND if (ARP_MAC_Source = IP_MAC_Mapping_Table [i,j] ) ;
    then Print ("This ARP packet is valid") ;
    else Print ("This packet is spoofed & rejected; Case #1") ;
} /* for detecting packet P#1 */

Cross_Layer_Inspection_ARP_Request (Ethernet_MAC_Source, ARP_MAC_Source)

{ for i > 0 , j > 0 , i ++ , j ++;
    if (ARP_IP_Source = IP_MAC_Mapping_Table [i,j]) ;
    else if (ARP_MAC_Source = Ethernet_MAC_Source [i,j] ) ;
}

```



```

        then Print ("This ARP packet is valid") ;
        else Print ("This packet is spoofed & rejected; Case #2") ;
    } /* for detecting packet P#2 */

Unicast_ARP_request (Ethernet_MAC_Destination);

{ for i > 0 , j > 0 , i ++ , j ++;
    if (Ethernet_MAC_Destination = "00-00-00-00-00-00") ; /* check for
broadcast*/
        then Print ("This ARP packet is valid") ;
        else Print ("This packet is spoofed & rejected; Case #3") ;
    } /* for detecting packet P#3 */

Unexpected_IP_MAC_Addresses_in_ARP_Request (Ethernet_MAC_Source,
Ethernet_MAC_Destination, ARP_IP_Source, ARP_MAC_Source, ARP_IP_Destination,
ARP_MAC_Destination);

{ for i > 0 , j > 0 , i ++ , j ++;
    if (Ethernet_MAC_Source = ARP_MAC_Source [i,j]) ;
    else if (Ethernet_MAC_Destination = ARP_MAC_Destination [i,j] ) ;
    else if (ARP_IP_Source != "0.0.0.0") ; /* IP_Source should not be broadcast */
    else if (ARP_IP_Source != "0.0.0.0") ; /* IP_Source should not be broadcast */
    else if (Ethernet_MAC_Destination = "00-00-00-00-00-00") ; /*check for
broadcast*/
        then Print ("This ARP packet is valid") ;
        else Print ("This packet is spoofed & rejected; Case #2") ;
    } /* for detecting packet P#4 */
}

Abnormal_Arp_Packet_Detection (Ethernet_header, ARP_header,
IP_MAC_Mapping_Table)

/* ARP reply packet */

if (ARP_Operation = "reply");
{
IP_to_MAC_Address_Mappings_ARP_Reply (ARP_IP_Source, ARP_MAC_Source,
ARP_IP_Destination, ARP_MAC_Destination, IP_MAC_Mapping_Table );
IP_to_MAC_Address_Mappings_ARP_Request (ARP_IP_Source, ARP_MAC_Source,
IP_MAC_Mapping_Table ) /* for detecting packets P#5 and P#7 */

    { for i > 0 , j > 0 , i ++ , j ++;
        if (ARP_IP_Source = IP_MAC_Mapping_Table [i,j]) ;
        AND if (ARP_MAC_Source = IP_MAC_Mapping_Table [i,j] ) ;
        then Print ("This ARP packet is valid") ;
        else Print ("This packet is spoofed & rejected; Case #5") ;
    } /* for detecting packet P#5 */

    { for i > 0 , j > 0 , i ++ , j ++;
        if (ARP_IP_Destination = IP_MAC_Mapping_Table [i,j]) ;
        AND if (ARP_MAC_Destination = IP_MAC_Mapping_Table [i,j] ) ;
        then Print ("This ARP packet is valid") ;
        else Print ("This packet is spoofed & rejected; Case #7") ;
    } /* for detecting packet P#7 */
}

```

```

Cross_Layer_Inspection_ARP_Reply (Ethernet_MAC_Source, ARP_MAC_Source,
Ethernet_MAC_Destination, ARP_MAC_Destination ) /* for detecting packets P#6 and P#8
*/

{ for i > 0 , j > 0 , i ++ , j ++;
    if (ARP_MAC_Source = IP_MAC_Mapping_Table [i,j] ) ;
    AND if (ARP_MAC_Source = Ethernet_MAC_Source [i,j] ) ;
    then Print ("This ARP packet is valid") ;
    else Print ("This packet is spoofed & rejected; Case #6") ;
} /* for detecting packet P#6 */

{ for i > 0 , j > 0 , i ++ , j ++;
    if (ARP_MAC_Destination = IP_MAC_Mapping_Table [i,j] ) ;
    AND if (ARP_MAC_Destination = Ethernet_MAC_Destination [i,j] ) ;
    then Print ("This ARP packet is valid") ;
    else Print ("This packet is spoofed & rejected; Case #6") ;
} /* for detecting packet P#8 */

Broadcast_ARP_reply (Ethernet_MAC_Destination) /* ARP reply should be Unicast */

{ for i > 0 , j > 0 , i ++ , j ++;
    if (Ethernet_MAC_Destination = "00-00-00-00-00-00") ; /* check for
broadcast*/
    then Print ("This packet is spoofed & rejected; Case #3") ;
    else Print ("This ARP packet is valid") ;
} /* for detecting packet P#9 */

Unexpected_IP_MAC_Addresses_in_ARP_Reply (Ethernet_MAC_Source,
Ethernet_MAC_Destination, ARP_IP_Source, ARP_MAC_Source, ARP_IP_Destination,
ARP_MAC_Destination); /* for detecting packet P#10 */
{ for i > 0 , j > 0 , i ++ , j ++;
    if (Ethernet_MAC_Source = ARP_MAC_Source [i,j] ) ;
    else if (Ethernet_MAC_Destination = ARP_MAC_Destination [i,j] ) ;
    else if (ARP_IP_Source != "0.0.0.0") ; /* IP_Source should not be broadcast */
    else if (ARP_IP_Source != "0.0.0.0") ; /* IP_Source should not be broadcast */
    else if (Ethernet_MAC_Destination != "00-00-00-00-00-00") ; /*check for
Unicast*/
    then Print ("This ARP packet is valid") ;
    else Print ("This packet is spoofed & rejected; Case #10") ;
} /* for detecting packet P#10 */

}
}

```

Fig. 3.8: Sample of C++ code to detect ARP abnormal packets

4.3 Testing and evaluation of the Algorithm

The code was tested in the LAB by building up a model to simulate the normal scenario for filtering any ARP spoofed packets in a network. It was implemented in order to discover or detect the 10 types of ARP abnormal (spoofed) packets {P#1, P#2....P#10} which was shown in Table 3.1 and Table 3.2.

As it was demonstrated in figure (3.9), we used the following items:

- Two host computers, one for sending the abnormal ARP packets (Tester), and the other for receiving the filtered ARP packets after being filtered (Target)
- A workstation PC with two Network-cards connecting the workstation with the two hosts; Tester (NIC#1 and Target (NIC#2). The workstation will be simulating a Network switch and filters the upcoming ARP traffic from the Tester-host. The proposed algorithm was implanted in the Workstation using C++ programming language and testing the input traffic coming from the abnormal traffic generator (Tester Host) via NIC#1, filtered by the algorithm, and forward the ONLY normal traffic to the destination (Target Host) via NIC#2.
- Packet Generator, in order to generate the 10 different types of the ARP abnormal packets and send them from the Tester-host to the Target-host. However we used a Software packet generator installed in the Tester-host to generate the required packets for easiness. This software is called: FramelP and is available as a Freeware[47].
- The proposed algorithm was implemented by an object-oriented language: C++ code and installed in the Workstation to perform the filtering over the ARP traffic

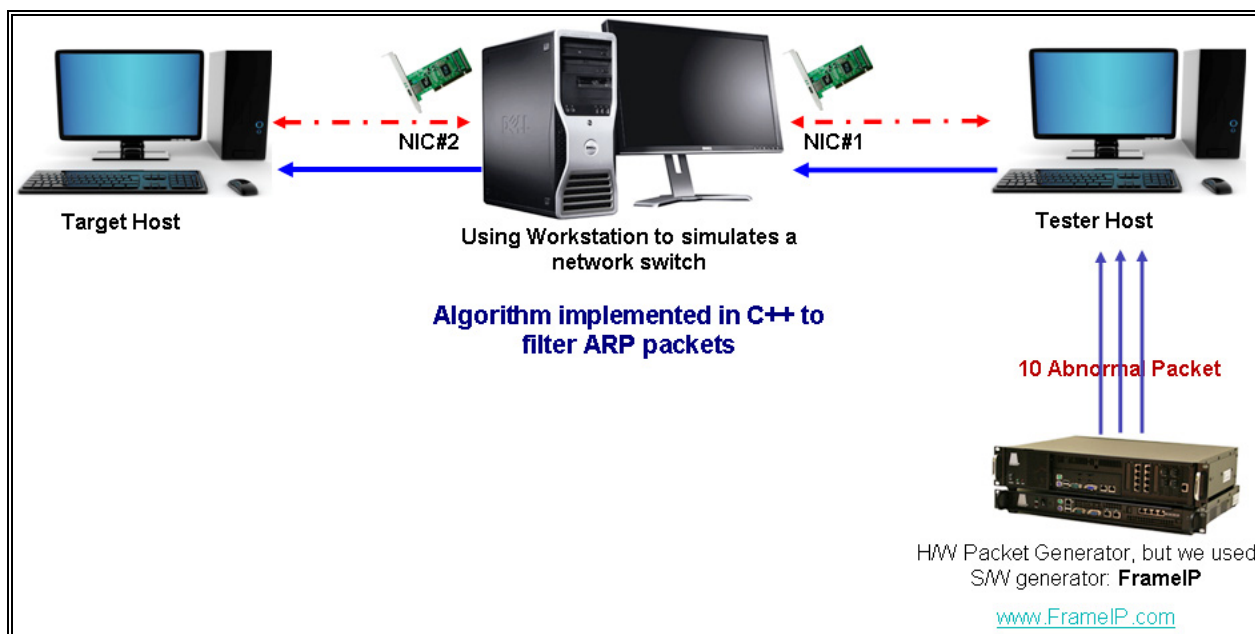


Fig. 3.9: The experiment compiled in the LAB to test the simulation switch algorithm

4.3.1 Results of the test experiment

The Algorithm was successful in filtering all so far known types of ARP spoofed packets (Abnormal). Only normal ARP traffic was forwarded to the destination (Target Host) and abnormal ARP packets were dropped from the traffic. On the other hand; the network performance was not considerably affected after applying the algorithm due to the reality that ARP packets have no data and are too small in size compared to TCP/IP packets.

4.3.2 Final Outcome of the Algorithms

Here we will try to provide a brief summary of the objectives and what are the best expectations while using the proposed Algorithm in this study and adopting it in network security devices such as Network Switches, Firewalls, ...etc. the following final outcome is written in short technical statements as follows:

1. Reject all types of known & identified malicious ARP traffic.
2. Deny malicious users from performing MiM attack based on scammed ARP packets.
3. Deny malicious users from performing DoS attack based on scammed ARP packets.

4. Protect LAN switches from MAC cloning attacks; this might occur when using RARP protocol to recall an IP address that correspond to a certain MAC address and when more than one IP address is returned, then this case is called MAC cloning.
5. Deny malicious users from performing attacks on LAN switches, mainly making the switch behaves as a broadcast switch (Hub).

4.4 Lab work

4.4.1 Packet Generation using FramelP Tool

The objective of this lab is to send multiple types of packets such as; ARP, IP, TCP, UDP and ICMP to a certain host. We'll create a fake (spoofed) and malicious traffic through the network that can be possibly carrying attacks. This will help us generally to test any security device or algorithm, such as Firewall, Intrusion Detection System [IDS], Intrusion Protection System [IPS], Firewall and even Anti-sniffer.

Initially, we need to download the FramelP from [47] to any folder in the PC then install the WinpCap from [48] which was used to capture the Packet and monitor the network, herewith the FramelP run in figure 4.4.1, however the FramelP commands and help is in the Appendix [2].

```
C:\Documents and Settings\Moh'd\Desktop\FramelP>frameip
FrameIP - Create some IP frame - Version 5.10.3.13
Create on December 21, 2002, Last compilation on June 02, 2009
Created by Sebastien FONTAINE - http://www.frameip.com

GENERAL OPTIONS
-?                This help
-wait             Wait after frame   Default: 1000 ms
-loops           Number of loops   Default: 1 (0=>no stop)
-send_mode       0=Socket 1=Libpcap Default: 1
-view           Show the answers  Default: 1
```

Fig. 4.4.1: The help menu for FramelP Tool

- We need also to use a sniffer tool installed on the source traffic generator, to verify that the generated traffic has been sent correctly. So we will use; *CommView* sniffer tool [49].

Now we will show an example of how an ARP request packet is generated using FramelP tool:

- First consider the following ARP request table 4.4.1:

Table 4.4.1: ARP request details

ARP header	
Operation = 1 (request)	
Source IP = 172.22.0.180	
Source MAC = 00-90-4B-89-AA-91	
Destination IP = 20.20.20.20	
Destination MAC = 00-00-00-00-00-00	
Ethernet header	
Source MAC = 00-90-4B-89-AA-91	
Destination MAC = FF.FF.FF.FF.FF.FF (Broadcast)	
Ethernet Type = 0x0806 (ARP)	

- So, if we want to send ARP request packet, we need to collect the MAC address of the destination IP address, and we will use the following FramIP command with help of Table 4.4.1:

C:>FramIP –interface 2 –MAC_Type 2054 –ARP_IP_Destination 20.20.20.20

Where, interface 2; is the Wireless LAN network.

- Now we will send an ARP reply packet to a destination IP, and provide the required MAC address and IP address:

Table 4.4.2: ARP reply details

ARP header	
Operation = 2 (reply)	
Source IP = 172.22.0.180	
Source MAC = 00-90-4B-89-AA-91	
Destination IP = 20.20.20.20	
Destination MAC = 00-17-A4-D7-D7-60	
Ethernet header	
Source MAC = 00-90-4B-89-AA-91	
Destination MAC = 00-17-A4-D7-D7-60 (Unicast)	
Ethernet Type = 0x0806 (ARP)	

- The above ARP reply packet (Table 4.4.2) can be implemented by the following FramIP command;

```
C:>FramIP -interface 2 -MAC_Type 2054 -ARP_OpCode 2 - IP_Destination  
20.20.20.20 -ARP_MAC_Source 00-90-4B-89-AA-91 -MAC_IP_Source  
172.22.0.180
```

- To send an abnormal ARP packet, we can for instance, use the broadcast MAC address for the source host in the reply ARP message, as follows;

```
C:>FramIP -interface 2 -send_mode 1-mac_type 2054 -arp_mac_source FF-FF-  
FF-FF-FF-FF -arp_ip_destination 20.20.20.20 -loops 0 -wait 0
```

P.S. where “FF-FF-FF-FF-FF-FF” is the broadcast MAC address, “-loops 0” means NO stop,
“-wait 0”; stand for don’t wait between sending two ARP packets. And for more details about FramelP commands you may refer to the Manual in Appendix[2].

4.4.2 Detecting Fake ARP packets using Snort IDS

Snort is an open source network Intrusion Prevention System [IPS] and Intrusion Detection System (IDS). [24].

It is one of the most famous adopted Intrusion Detection System [IDS] and Intrusion Protection System [IPS] in Information Security area nowadays, since it combines the signature, protocol and anomaly-based inspection [24]. The good to mention Snort became the de-facto standard for intrusion prevention system because of the huge number of download and registered users.

Now we are going to demonstrate the second lab about: “Detecting fake ARP packets using Snort IDS”, the aim of this lab is to use Snort IDS to detect spoofed ARP packets, which may be used to perform MiM attack or DoS attacks.

The Snort IDS S/W should be installed on a machine which is must be connected to a SPAN port on the switch (as shown in figure 4.4.2.1). We will try to send about 6 types of fake ARP packets from the source (Host A) as broadcast or unicast through the network and Host C where the Snort IDS is installed will be tested against detecting the fake packets.

There are 3 hosts which are connecting a switch (A, B and C). and Host “C” is connected to Span port, as in the following configuration:

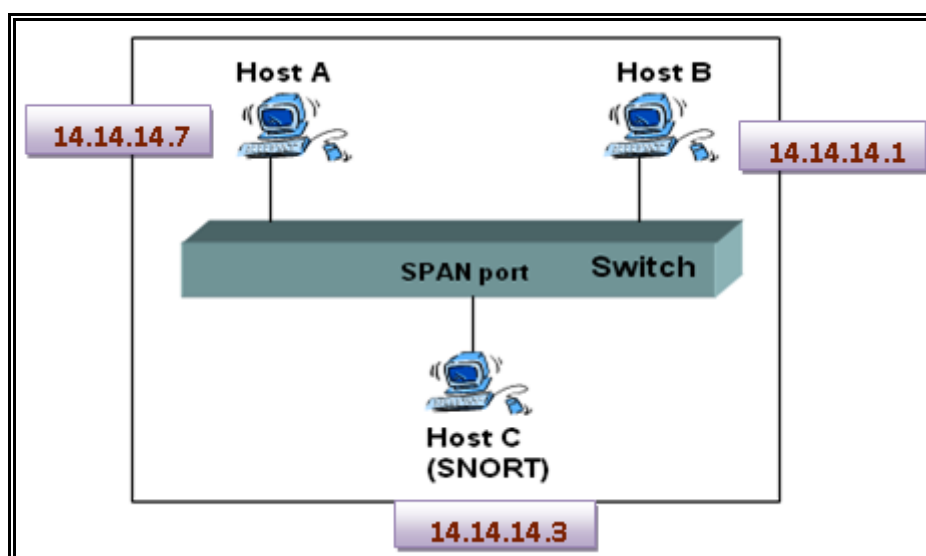


Fig. 4.4.2.1: Using Snort IDS to detect spoofed ARP packets

Configuration Phase:

1. Install Snort and WinPcap on host C
2. Unzip the file "rules" in the folder: "C:\Snort\rules"
3. Create the folder: "C:\Snort\Preproc_rules", and put the file: "Preprocessor.rules" in the same folder
4. Configure the file "Snort.Conf" which is located in "C:\Snort\etc\".
5. Specify the correct mapping between [IP & MAC] addresses for a hosts connected to the switch within the "Snort.cnfig" file as shown below

```

# SID      Event description
# -----  -
# 1        Unicast ARP request
# 2        Etherframe ARP mismatch (src)
# 3        Etherframe ARP mismatch (dst)
# 4        ARP cache overwrite attack

preprocessor arpspoof: -unicast
preprocessor arpspoof_detect_host: 14.14.14.7 00:11:43:4c:fe:f2 → Host A
preprocessor arpspoof_detect_host: 14.14.14.1 00:12:3f:17:56:e3 → Host B
preprocessor arpspoof_detect_host: 14.14.14.3 00:12:3f:18:6a:83 → Host C

```

6. Identify the interface and run Snort as a sniffer and as IDs.
7. Send Fake ARP packets and check the detection on the log and we use CommView sniffer to see the captured traffic through the SPAN port.

Packet#1: (Sender: Host A)

ARP operation	Request
Source IP	A
Source MAC	A
Destination IP	C
Destination MAC	
Destination MAC	C
Source MAC	A
MAC type (IP or ARP)	ARP

It is an ARP request packet sent by the host A to the host C, The problem with this packet is that it's a request's unicast packet

We sent this packet from Host A which is an ARP request to C MAC destination "Unicast", which is abnormal because it must be broadcast.

```

\FrameIP>frameip -interface 1 -send mode 1 -loops 1 -mac_tune 2054 -arp_opcode 1
-mac_source 00-11-43-4c-fe-f2 -mac_destination 00-12-3f-18-6a-83 -arp_mac_source
e 00-11-43-4c-fe-f2 -arp_ip_destination 14.14.14.3

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
  
```

Unicast MAC

Detecting Result:

This is the log file shows that the snort detects the above fake ARP packet

```

[**] [112:1:1] (spp_arpspoof) Unicast ARP request [**]
04/08-13:28:24.125364

[**] [112:1:1] (spp_arpspoof) Unicast ARP request [**]
04/08-13:28:25.122155

[**] [112:1:1] (spp_arpspoof) Unicast ARP request [**]
04/08-13:28:26.122139
  
```

1 means that there is Unicast ARP request

This packet captured by host C which shows the following:

No	Protocol	MAC Addresses	IP Addresses
1	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
2	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
3	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
4	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
5	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
6	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
7	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
8	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
9	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
10	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
11	IP/UDP	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.7 <=> 14.14.14.255
12	IP/UDP	00:12:3F:18:6A:83 => Broadcast	14.14.14.3 => 14.14.14.255
13	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3
14	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -! 14.14.14.7
15	ARP REQ	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -? 14.14.14.3

Packet#2: (Sender: Host A)

ARP operation	Request
Source IP	A
Source MAC	B
Destination IP	C
Destination MAC	0
Destination MAC	BROUDCAST
Source MAC	A
MAC type (IP or ARP)	ARP

Fake ARP request packet sent by the host A to the host C, claiming that he is B host. So If host C send to B, it will be directed to Host A

We sent a request packet from Host A to Host C, claiming that the sender is B, to assign the A's MAC to B's IP, so later if the Host C send packets to B, it will be forwarded to host A not B, and that is abnormal.

```

FrameIP>frameip -interface 1 -loops 0 -send mode 1 -mac type 2054 -arp_opcode 1
-arp_ip_source 14.14.14.1 -arp_mac_source 00-11-43-4c-fe-f2 -arp_ip_destinatio
n 14.14.14.3

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-11-43-4C-FE-F2 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to FF-FF-FF-FF-FF-FF with 57 Bytes
  
```

Detecting Result:

This is the log file shows that the snort detects the above fake ARP packet

```

[**] [112:4:1] (spp_arp spoof) Attempted ARP cache overwrite attack [**]
04/08-13:35:12.649873

[**] [112:4:1] (spp_arp spoof) Attempted ARP cache overwrite attack [**]
04/08-13:35:13.649330

[**] [112:4:1] (spp_arp spoof) Attempted ARP cache overwrite attack [**]
04/08-13:35:14.649881

[**] [112:4:1] (spp_arp spoof) Attempted ARP cache overwrite attack [**]
04/08-13:35:15.649376

```

4 means that there is a mismatch in the MAC destination

This packet captured by host C which shows the following:

Ethernet II		No	Protocol	MAC Addresses	IP Addresses
Destination MAC: FF:FF:FF:FF:FF:FF		1	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Source MAC: 00:11:43:4C:FE:F2		2	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Ethertype: 0x0806 (2054) - ARP		3	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Direction: Pass-through		4	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Time / Delta Time: 15:01:37.872 / 75		5	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Frame size: 60 bytes		6	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Frame number: 1		7	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
ARP		8	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Hardware: 0x0001 (1) - Ethernet		9	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Protocol: 0x0800 (2048) - IP		10	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Hardware address length: 0x06 (6)		11	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Protocol address length: 0x04 (4)		12	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Operation: 0x0001 (1) - ARP Request		13	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Sender MAC address: 00:11:43:4C:FE:F2		14	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Sender IP address: 14.14.14.1		15	ARP REQ	00:11:43:4C:FE:F2 <=> Broadcast	14.14.14.1 -> 14.14.14.3
Target MAC address: 00:00:00:00:00:00		16	ARP RESP	00:12:3F:18:6A:83 => 00:11:43:4C:FE:F2	14.14.14.3 -> 14.14.14.1
Target IP address: 14.14.14.3					

A's MAC address

B's IP address

Packet#3: (Sender: Host B)

ARP operation	Reply
Source IP	B
Source MAC	B
Destination IP	A
Destination MAC	A
Destination MAC	BROADCAST
Source MAC	B
MAC type (IP or ARP)	ARP

It is an ARP reply packet so this should be unicast (MAC of A), not a broadcast

We sent this packet from Host B which is an ARP reply with the broadcast MAC destination, which is abnormal, and there is a mismatch in the destination MAC address between an Ethernet layer and an ARP layer[53]:

```

der\Packet generator frame IP\Console version>frameip -interface 1 -loops 0 -sen
d_mode 1 -mac_type 2054 -arp_opcode 2 -mac_source 00-12-3f-17-56-e3 -mac_destina
tion ff-ff-ff-ff-ff-ff -arp_ip_source 14.14.14.1 -arp_mac_source 00-12-3f-17-56-
e3 -arp_ip_destination 14.14.14.7 -arp_mac_destination 00-11-43-4c-fe-f2

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-12-3F-17-56-E3 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-12-3F-17-56-E3 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-12-3F-17-56-E3 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-12-3F-17-56-E3 to FF-FF-FF-FF-FF-FF with 57 Bytes
The frame was sent from 00-12-3F-17-56-E3 to FF-FF-FF-FF-FF-FF with 57 Bytes
  
```

Detecting Result:

This is the log file shows that the snort detects the above fake ARP packet:

```

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:44:33.502328

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:44:34.501606

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:44:35.501764
  
```

3 means that there is a mismatch in the MAC destination

This packet captured by host C which shows the following:

The image shows a Wireshark packet capture analysis. On the left, the Ethernet II layer details are shown: Destination MAC: FF:FF:FF:FF:FF:FF, Source MAC: 00:12:3F:17:56:E3, Ethertype: 0x0806 (2054) - ARP, Direction: Pass-through, Time / Delta Time: 18:49:38.781 / 0.000, Frame size: 57 bytes, Frame number: 1. Below this, the ARP layer details are shown: Hardware: 0x0001 (1) - Ethernet, Protocol: 0x0800 (2048) - IP, Hardware address length: 0x06 (6), Protocol address length: 0x04 (4), Operation: 0x0002 (2) - ARP Response, Sender MAC address: 00:12:3F:17:56:E3, Sender IP address: 14.14.14.1, Target MAC address: 00:11:43:4C:FE:F2, Target IP address: 14.14.14.7. A callout box points to the Destination MAC in the Ethernet II layer and the Target MAC address in the ARP layer, stating: "Mismatch in the MAC address of the target, and ARP reply can not be broadcast". On the right, a packet list table shows several ARP RESP packets with broadcast MAC addresses and IP addresses 14.14.14.1 and 14.14.14.7. Below the table, a hex dump shows the raw packet data.

No	Protocol	MAC Addresses	IP Addresses
4	ARP RESP	00:12:3F:17:56:E3 <=> Broadcast	14.14.14.1 -> 14.14.14.7
2	ARP RESP	00:12:3F:17:56:E3 <=> Broadcast	14.14.14.1 -> 14.14.14.7
1	ARP RESP	00:12:3F:17:56:E3 <=> Broadcast	14.14.14.1 -> 14.14.14.7
8	ARP RESP	00:12:3F:17:56:E3 <=> Broadcast	14.14.14.1 -> 14.14.14.7
3	ARP RESP	00:12:3F:17:56:E3 <=> Broadcast	14.14.14.1 -> 14.14.14.7

Packet#4: (Sender: Host C)

ARP operation	Request
Source IP	A
Source MAC	A
Destination IP	B
Destination MAC	0
Destination MAC	BROADCAST
Source MAC	C
MAC type (IP or ARP)	ARP

It is an ARP request packet from host C, but it shows in the ARP layer that the sender is host A, so its overwrite attack

We sent this packet from Host C which is an ARP request but it shows in the ARP layer that it's from A

```
der\Packet generator frame IP\Console version>frameip -interface 2 -send_mode 1  
-loops 0 -mac_type 2054 -arp_opcode 1 -mac_source 00-12-3f-18-6a-83 -arp_ip_sour  
ce 14.14.14.7 -arp_mac_source 00-11-43-4c-fe-f2 -arp_ip_destination 14.14.14.1  
FrameIP - Create some IP frame - Version 5.9.3.12  
Create on December 21, 2002. Last compilation on March 13, 2007  
Created by Sebastien FONTAINE - http://www.frameip.com  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes  
The frame was sent from 00-12-3F-18-6A-83 to FF-FF-FF-FF-FF-FF with 57 Bytes
```

Detecting Result:

This is the log file shows that the snort detects the above fake ARP packet

```
[**] [112.4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]  
04/08-13:55:49.242180  
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]  
04/08-13:55:50.242203  
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]  
04/08-13:55:51.242226  
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]  
04/08-13:55:52.242276
```

4 means that there is an ARP cache overwrite attack.

This packet captured by host C which shows the following:

No	Protocol	MAC Addresses	IP Addresses
1	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
2	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
3	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
4	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
5	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
6	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
7	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
8	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1
9	ARP REQ	00:12:3F:18:6A:83 => Broadcast	14.14.14.7 -> 14.14.14.1

Packet#5: (Sender: Host A)

ARP operation	Reply
Source IP	A
Source MAC	A
Destination IP	C
Destination MAC	B
Destination MAC	C
Source MAC	A
MAC type (IP or ARP)	ARP

It is an ARP reply packet sent by the host A to the host C, but it shows on the ARP layer the it sent to the host B, so its Ethernet ARP destination MAC mismatch

We sent this packet from Host A to Host C, but there is mismatch on the destination MAC address of ARP layer and Ethernet layer which is abnormal packet


```

NFrameIP>frameip -interface 1 -loops 5 -send_mode 1 -mac_type 2054 -arp_opcode 2
-mac_source 00-11-43-4c-fe-f2 -mac_destination 00-12-3f-18-6a-83 -arp_ip_source 1
4.14.14.7 -arp_mac_source 00-11-43-4c-fe-f2 -arp_ip_destination 14.14.14.3 -arp
mac_destination 00-12-3f-17-56-e3

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-18-6A-83 with 57 Bytes

```

Detecting Result:

This is the log file shows that the snort detects the above fake ARP packet

```

[**] [112:3:1] (spp_arp spoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:35:06.561321

[**] [112:3:1] (spp_arp spoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:35:07.552925

[**] [112:3:1] (spp_arp spoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:35:08.554596

[**] [112:3:1] (spp_arp spoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:35:09.552745

```

3 means that there is Ethernet ARP mismatch (Destination MAC Address)

This packet captured by host C which shows the following:

Ethernet II		No	Protocol	MAC Addresses	IP Addresses
Destination MAC: 00:12:3F:18:6A:83		1	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Source MAC: 00:11:43:4C:FE:F2		2	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Ethertype: 0x0806 (2054) - ARP		3	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Direction: In		4	ARP	00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Time / Delta Time: 14:35:06.559 / 0.000		5	ARP	00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Frame size: 60 bytes		6	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Frame number: 1		7	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
ARP		8	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Hardware: 0x0001 (1) - Ethernet		9	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Protocol: 0x0800 (2048) - IP		10	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Hardware address length: 0x06 (6)		11	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Protocol address length: 0x04 (4)		12	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Operation: 0x0002 (2) - ARP Response		13	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Sender MAC address: 00:11:43:4C:FE:F2		14	ARP	00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Sender IP address: 14.14.14.7		15	ARP	00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Target MAC address: 00:12:3F:17:56:E3		16	ARP RESP	00:12:3F:18:6A:83 <= 00:11:43:4C:FE:F2	14.14.14.7 -> 14.14.14.3
Target IP address: 14.14.14.3					

C's MAC address

B's MAC address

Packet#6: (Sender: Host A)

ARP operation	Reply
Source IP	A
Source MAC	A
Destination IP	C
Destination MAC	C
Destination MAC	B
Source MAC	A
MAC type (IP or ARP)	ARP

It is an ARP reply packet sent by the host A to the host C, but it shows on the Ethernet layer that it was sent to the host B, so their Ethernet ARP destinations MAC mismatch

We sent this packet from Host A to Host C, but there is mismatch on the destination MAC address of ARP layer and Ethernet layer which is abnormal packet

```
\FrameIP>frameip -interface 1 -loops 0 -send_mode 1 -mac_type 2054 -arp_opcode 2
-mac_source 00-11-43-4c-fe-f2 -mac_destination 00-12-3f-17-56-e3 arp_ip_source 1
4.14.14.7 -arp_mac_source 00-11-43-4c-fe-f2 -arp_ip_destination 14.14.14.3 -arp
mac_destination 00-12-3f-18-6a-83

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
The frame was sent from 00-11-43-4C-FE-F2 to 00-12-3F-17-56-E3 with 57 Bytes
```

Detecting Result:

This is the log file which shows that the snort detects the above fake ARP packet

```
[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:38:58.496859

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:38:59.495424

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:39:00.495325

[**] [112:3:1] (spp_arpspoof) Ethernet/ARP Mismatch request for Destination [**]
04/08-14:39:01.495333
```

This packet captured by host C which shows the following:

Ethernet II		No	Protocol	MAC Addresses	IP Addresses
Destination MAC: 00:12:3F:17:56:E3		1	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Source MAC: 00:11:43:4C:FE:F2		2	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Ethertype: 0x0806 (2054) - ARP		3	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Direction: Pass-through		4	ARP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Time / Delta Time: 14:38:58.497 / 131.188		5	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Frame size: 60 bytes		6	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Frame number: 1		7	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
ARP		8	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Hardware: 0x0001 (1) - Ethernet		9	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Protocol: 0x0800 (2048) - IP		10	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Hardware address length: 0x06 (6)		11	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Protocol address length: 0x04 (4)		12	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Operation: 0x0002 (2) - ARP Response		13	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Sender MAC address: 00:11:43:4C:FE:F2		14	ARP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Sender IP address: 14.14.14.7		15	ARP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Target MAC address: 00:12:3F:18:6A:83		16	ARP RESP	00:11:43:4C:FE:F2 <=> 00:12:3F:17:56:E3	14.14.14.7 -> 14.14.14.3
Target IP address: 14.14.14.3					

CHAPTER 5

Effect of Common Network Attacks on Biometrics performance

5.1 Biometric technologies

Biometric technologies may seem exotic, but their use is becoming increasingly common [55]. Biometrics Readers and Scanners are being used and implemented in almost everywhere nowadays, such as government places, schools, universities, hospitals, military basis and army...etc. they are mainly used for user identification and verification. They play an important role in implementing the security policies within the institutions. Most biometrics readers are able to connect to local area networks (LAN), and communicate with remote biometric servers to exchange biometric data. Figure below shows Biometrics types;

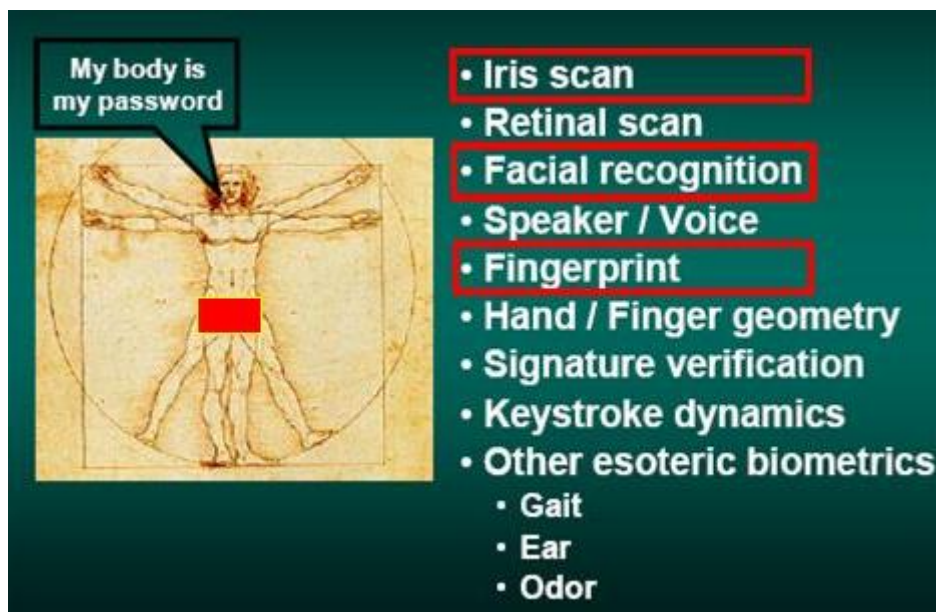


Fig. 5.1: Biometrics as; “Leonardo DaVinci’s Vitruvian Man” [55]

In the above figure: 5.1; we can see the different types of Biometrics for Humans; these may include; Iris scan, Facial recognition, Fingerprint, Signature and some other types...etc. In our research we will apply the attacking technique against: Fingerprint and iRIS scanners only and evaluate their performance against network threads.

5.1.1 iRIS Scan:

Iris scanning measures the iris pattern in the colored part of the eye [55], it is a quick technique for identification and verification, for the reason that iRIS is private and reliable [55].

5.1.2 Fingerprint:

In the past, human used the old fashion technique which was called “ink-and-paper” to stamp a person’s fingerprint to identify his or her identity. This was also used for more than one hundred years. Now the human fingerprint is scanned and read electronically and converted digitally [55].

5.1.3 Facial Recognition:

Face recognition trace the geometrical spaces and remarkable facial appearance of the human face. There are many techniques for face recognition available in the market nowadays, based on measuring the key features of the face [55].

In this chapter we will investigate the effect of common network attacks on the performance of several fingerprint and iris readers. Experiments are conducted using Denial of Service attacks (DoSs), ARP cache poisoning attack and some other attacks. The experiments show clearly that the tested biometric readers are very vulnerable to DoS attacks, and their recognition performances are significantly affected after launching the attacks. However, the experiments confirm that the tested biometric readers are secure from ARP cache poisoning attack due to significant reasons which we are going to discuss in this chapter. It demonstrates also that biometric readers are easy targets for malicious network users since they lack security solutions, such as Firewalls, and Intrusion Detection/Prevention Systems.

The Security experts and Biologist presented the Biometrics technology as means of; providing the right person with the right privileges the right access at the right time [55]. However, the security level for the Biometrics is becoming questionable after we discovered and demonstrated in this research that most of the Biometrics needs to perform a standpoint and a whole review on the protection of the Biometrics against famous and vital security flaws..!

Biometric reader manufacturers have been focusing on offering an easy to use and practical device with effective cost, low enrollment and recognition time, and low rate of false match and non-match. However, since these devices are as any network host with IP and MAC addresses, they may be the targets of malicious network users.

In this paper we investigated the effect of common network attacks on the performance of several fingerprint and iRIS readers. Experiments are conducted by using Denial of Service attacks (DoSs) and ARP cache poisoning attack. And it shows clearly that the tested biometric readers are very vulnerable to most DoS attacks. Their performances are significantly affected by DoS attacks. Biometric readers lack robust security solutions, such as Firewalls or Intrusion Detection/Prevention systems, and consequently are easy targets for malicious network users.

The remainder of this paper is organized as follows. We'll present an overview about what's Biometrics and how it's work. Then we'll discuss the network attacks that have been used in the experiments. Then we'll present and analyze the experiments results and finally the conclusion.

5.1.4 What is Biometrics and how it works:

In 2001 MIT Technology Review [56] named biometrics as one of the “top ten emerging technologies that will change the world”. The term “Biometric” comes from the Greek words “bio” which means (life) and “metric” which means (to measure)[57]. A biometrics technology is being used to know someone's unique identity. Biometrics has two kinds: The Physical and the Behavioral. Physical Biometrics is usually used for Identification or Verification. And the Behavioral Biometrics that can be used for verification [57] .

Identification determines the identity of a person. It usually uses a database of Biometrics data for users' biometrics records and starts finding the matching records

and hence; identifies the user. However such process needs time and huge processing consuming when the compared database is too large..!

Verification determines if the user identity is valid and if he is telling the truth about his or her identity.

It usually use the user biometrics data which was collected formerly and compare it with the current user's data in order to guarantee that he or she is the identical individual. The verification process is less expensive than the identification process in terms of time for processing and power consuming. It is also used for authenticating users and controlling their access.[57]

Physical Biometrics Technology is divided mainly into different categories, it is including; Iris scan, Facial recognition, Fingerprint, Signature and some other types...etc. In our research we will apply the attacking technique against: Fingerprint and iRIS scanners only and evaluate their performance against network threads [56] and [57]. There are some other Biometrics categories which are still in their investigation range such as; DNA Biometrics, Fingernails and Body Odor [58]. However, they seem to be promising results in the future.

On the other hand, primarily and recognizable biometrics types are the Fingerprint, iRIS, Voice-Recognition, Hand-Geometry, and Face-Recognition, according to: [58],[59],[60], and [61]. However, depending on the application the Biometrics is used, it determines its principle and usage, and each type of the Biometrics types we just mentioned has its advantages and disadvantages.

Currently, Fingerprint and iRIS Biometrics are commonly used because they are express consistent, secure, not expensive, and grant outstanding recognition accuracy ratio [62]. According to a conducted study in 2004 [63] iRIS Biometrics is the most accurate Biometric Identification method among all other Biometrics techniques. Its false acceptance rate is too small, which means an invalid recognition possibility for a person as another person is almost zero [63].

For these reasons, we'll focus here while investigating the effect of common network attacks on Biometrics on the most two and common types: The Fingerprint and iRIS Readers, as a real application affecting our day-to-day life.

5.1.5 Network Attacks against Biometrics

As we mentioned prior in this study; Denial-of-Service [DoS] and ARP Cache Spoofing attacks are our focus in evaluating the security protection of the Biometric devices.

In particular, we conducted two major experiments, initially we; investigate the effect of DoS attacks on the performance of fingerprint and iRIS readers. However, in the following experiment, we investigate the ability of ARP cache poisoning attack to corrupt the ARP cache entries of the Biometric readers. In fact, network hosts with corrupted ARP caches may not be able to communicate appropriately with the other hosts in the network.

DoS Attacks & Biometrics:

Denial-of-Service [DoS] attack occur when any suspicious host over the network performs ARP cache poisoning and receives any packet designated to the original target to the suspicious host and cause a block in the connection between the host and the target which is being attacked. It also causes a slow down the application access which is hosted on the Servers and overloads the system. It may attack the entire enterprise and block the outgoing and incoming traffic to the host services.

The DoS attacks is spreading out everywhere nowadays, this is due to the reason that it's easy to conduct compared to the other attacks which require highly administrative privileges.

While, most of the DoS attacks are intentional attacks, where, the attacker deliberately plans to overload the server resources. On the other hand, the user may cause unplanned load on the server and hence, make it occupied accidentally.

Most DoS attacks depend on the limitation in the TCP/IP protocols. In the next section we will introduce some DoS attacks that were used in our experiments, specifically the SYN Flood, Land attack, TearDrop and UDP Flood attacks.

5.1.6 Land Attack:

Land attack takes place when the suspicious host or attacker is sending Poisoned [TCP-SYN] packets. The packet should include; the destination host IP-Address as well as the port used for the Source & Destination.

After that, the destination host with reply by: [SYN-ACK] packet, generating an empty connection staying up to the timeout.

Such type of system flooding by empty connections may cause system crash and then a Denial-of-Service. As illustrated in Figure 5.2

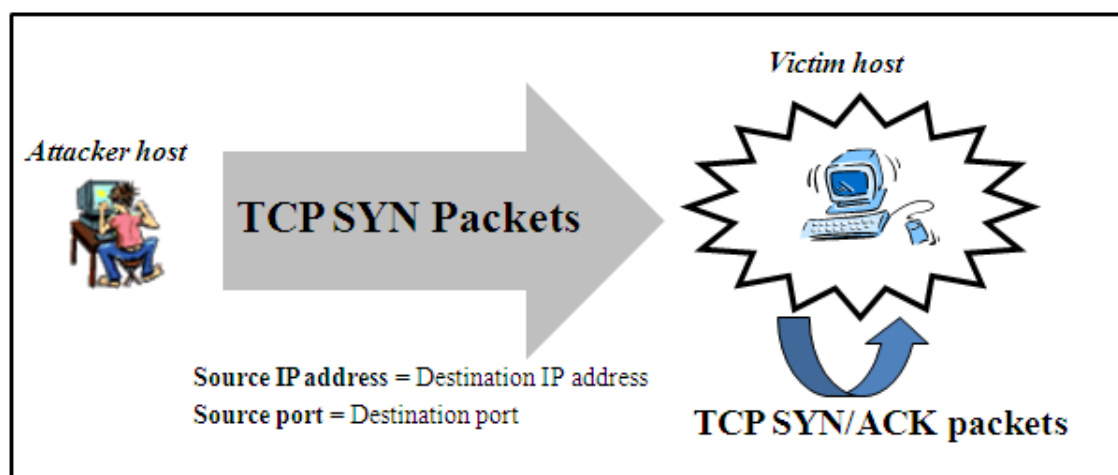


Fig. 5.2: The Land attack

5.1.7 SYN Flood Attack:

A SYN flood [Synchronization] starts when overloading hosts by [SYN Packets] and starting uncompleted requests for connection with host which it never be finished.

When the SYN message is sent from the source host to the host server, it replies by a SYN-ACK packet (Synchronization-Acknowledgment) back to the source host. Later a connection is established between the host and the server.

The real problem is in the case for Half-Opened-Connection. It occurs when the server sent [SYN-ACK] message and doesn't receive the final ACK message from the host.

Any system has a limited memory for the connections details with certain information about the connection such as Data Size and Data Structure. So creating so many numbers of Half-Opened connections may cause system overflow. As illustrated in Figure 5.3

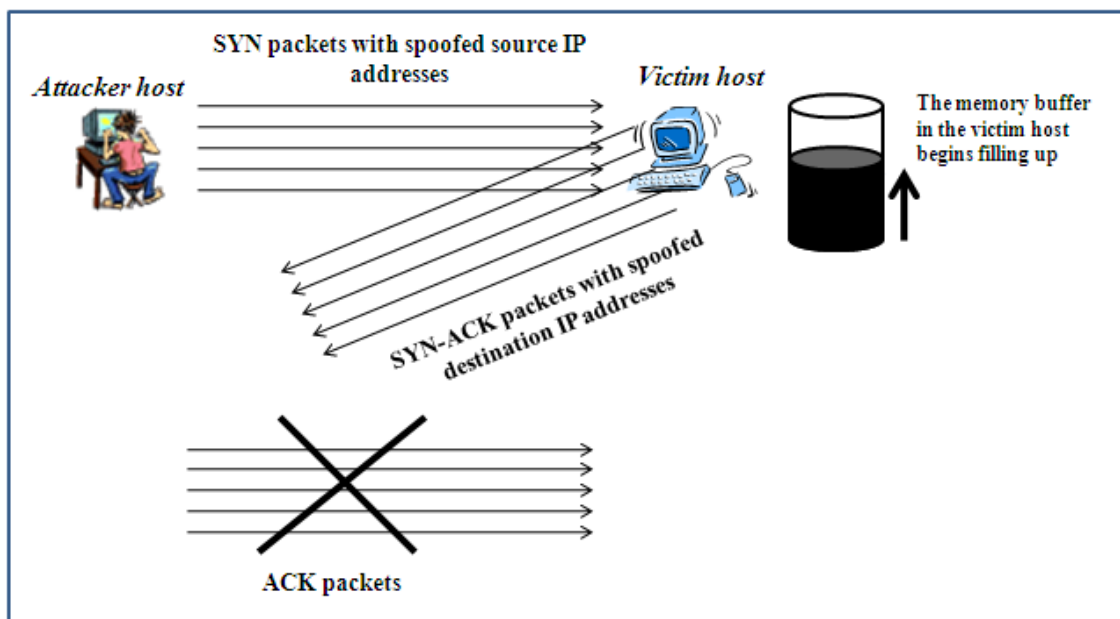


Fig. 5.3: The SYN Flood attack

When a numerous number of Half-Opened connections is generated, which is also an easy process using Spoofed IP packets, the malicious attacker is sending SYN requests to the target host as a valid connections, however the source IP address is invalid and doesn't show the source ID and hence this will be very tricky to trace and detect.

5.1.8 Teardrop Attack:

Teardrop attack is vulnerable because it fragments IP packets. When, the IP datagram is longer than the max. Transmission unit (MUT) of a network segment, the fragmentation is essential.

For the Teardrop attack the fragmented packets are intentionally formulated with overlapping offset fields which may cause the host to crash while trying to reassemble the packets.

In the following Figure 5.4 illustrates the Teardrop attacks

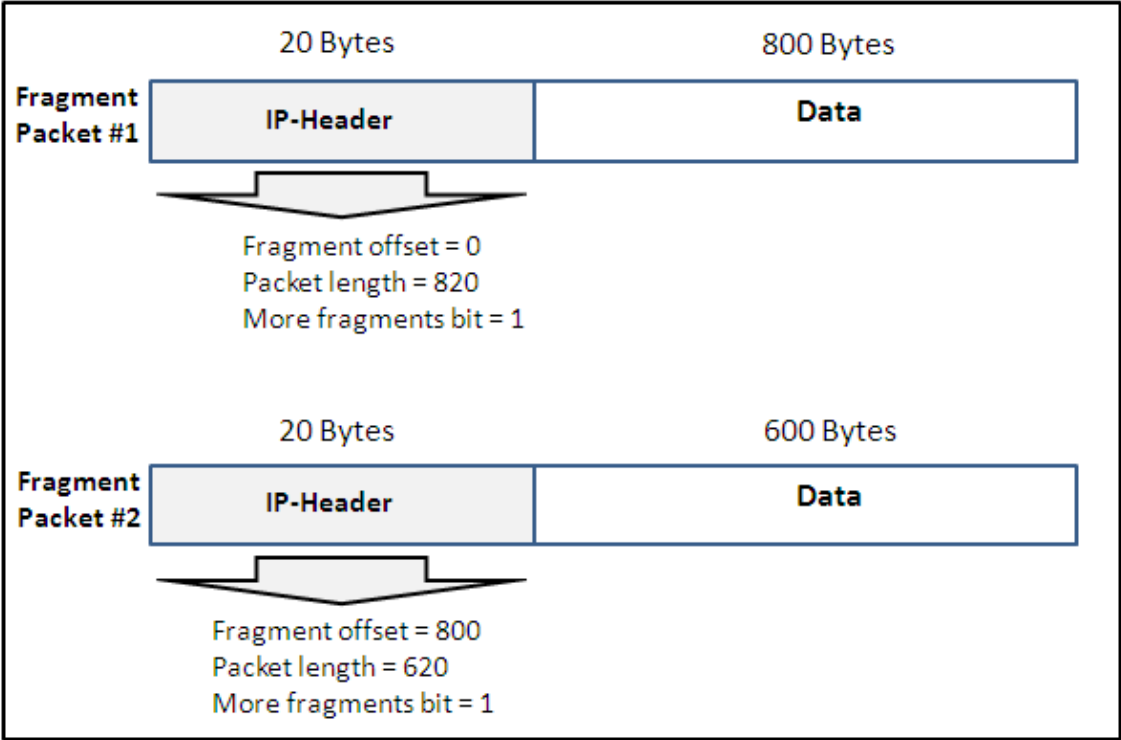


Fig. 5.4: The Teardrop attack

5.1.9 UDP Flood Attack:

UDP Flood Attack UDP is a connectionless protocol and it does not require any connection setup procedure to transfer data [64].

UDP Flood attack, taking place when the malicious host is start sending UDP packets to the target host's ports randomly.

This will decide which application will be holding over the UDP port. Then either the UDP port will be closed or return a ICMP message that the target host is not reachable.

As a result to the above action, the attacked system or host will face a network slowness or timeout. Figure 5.5 shows how UDP Flood is occurring.

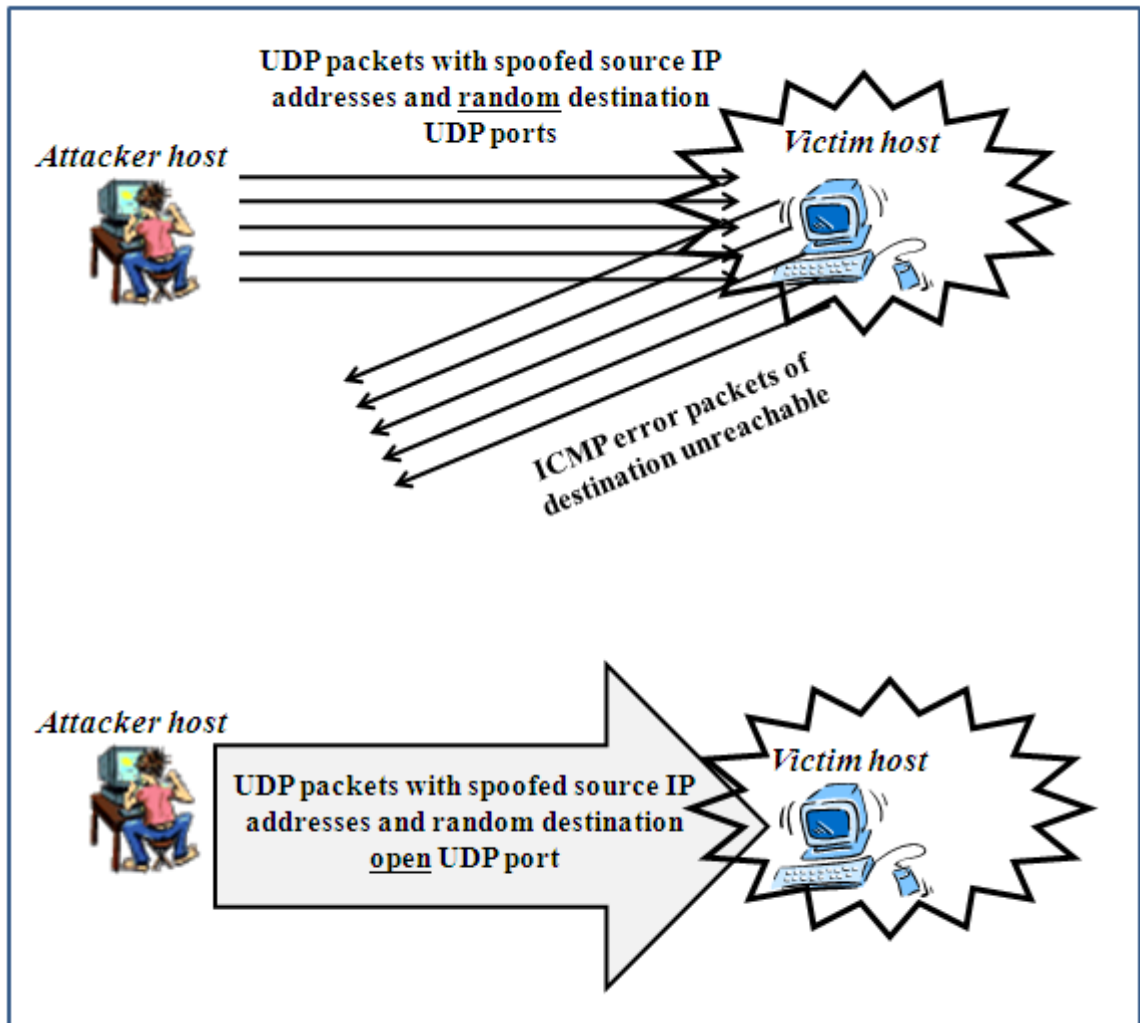


Fig. 5.5: UDP Flood attack

5.1.10 ARP Cache Poisoning effect on Biometrics

On shared broadcast LAN networks, packets sniffing can easily be achieved. However, a switched LAN has a problem called sniffing. It redirects the network traffic to a suspicious destination instead of the two source and target hosts. This will interrupt the communications between the designated hosts and even without their knowledge [16].

MiM attack is commonly used for network sniffing. It uses ARP Packet Spoofing to attack the target hosts [16]. It initiates a fake IP and MAC addresses match and corrupt the host ARP cache tables. Then the invalid ARP entry will be added to the ARP cache entries or it updates an existing ARP entry.

MiM attacker enables the IP Packet routing between his machine and the bombarded hosts. This is because the attacker doesn't want them to feel any change in their connection or a Denial of Service [DoS]. After that the two infected hosts keep forwarding their packets to the malicious host. As we mentioned earlier, if the IP Packet routing is not enabled by the attacker, then a Denial of Service [DoS] will immediately taking place. Figure 5.6 show how Biometric data are been sniffing based on the MiM attack.



Fig. 5.6: Biometric data sniffing based on the MiM attack

5.1.11 Experiments: Common network attacks vs. Biometric devices

In this section we conducted two experiments. In the first experiment, we investigate the effect of four common DoS attacks on the performance of several fingerprint and iris readers. In the second experiment, we examined the effect of ARP Packet Spoofing attack on ARP cache entries for several Biometric Readers and Scanners.

Network Architecture

In the following figure 5.7, we are showing the network architecture we used in order to conduct the experiments. Three attacker hosts, a biometric server, and fingerprint and iRIS readers are connected to a switch. The attacks are launched from the three attack hosts using two tools, which we will explain in the next section.

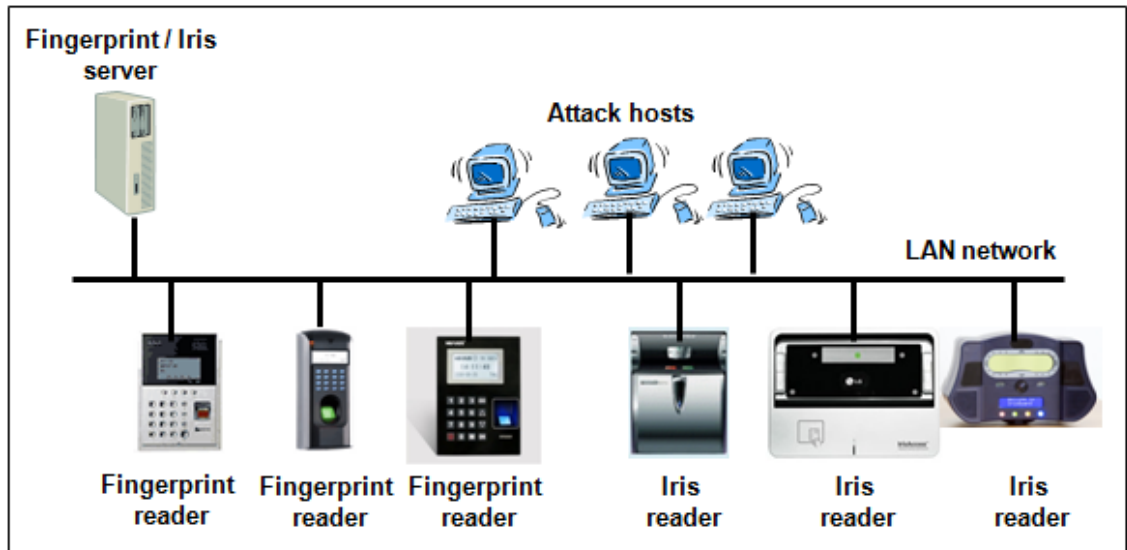


Fig. 5.7: Network architecture for Biometric Readers & iRIS scanners attacks

5.1.12 Attacking Tools against Biometrics

We used the following two tools to execute the experiments:

1. FramelIP packet generator [47] is a packet generator that allows generating any type of IP and ARP packets. The tool is used by the attacker hosts to generate the Land attack, Teardrop attack, and UDP Flood attacks; and it also used to perform ARP Cache Poisoning attack.
2. SYNflood tool [51] is a ready-to-use attack tool used to generate the SYN flood attack

Figure 5.8 show the command line used to generate the SYN flood attack, using the “SYN-Flood” tool. After executing SYN-Flood, a flood of fake TCP SYN packets are sent to the target Biometric reader whose IP address is 10.10.10.5 as shown.

```

C:\WINDOWS\system32\cmd.exe
D:\Tools\SYN Flood>
D:\Tools\SYN Flood>
D:\Tools\SYN Flood>
D:\Tools\SYN Flood>synflood -interface 2 -ip_destination 10.10.10.5 -loops 0

SynFlood - Send SYN TCP with spoofing IP Source - Version 2.3.3.10
Create on July 23, 2003, Last compilation on August 26, 2006
Created by sebastien.fontaine@frameip.com

The SYN TCP was sent from 115.3.45.144:35118 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 160.67.147.106:42363 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 216.103.64.150:11301 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 74.98.223.142:26976 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 138.247.162.64:47995 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 15.64.188.143:5503 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 181.43.231.208:32284 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 60.247.222.78:37232 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 199.188.228.199:65298 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 93.224.60.237:2167 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 108.163.175.230:1564 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 65.148.9.204:57869 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 145.25.156.45:39989 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 243.235.188.140:51321 to 10.10.10.5:80 - 69 Bytes
The SYN TCP was sent from 99.152.65.223:2863 to 10.10.10.5:80 - 69 Bytes

```

Fig. 5.8: The SYN flood attack online command

All four DoS attacks (SYN flood, Land, Teardrop and UDP flood attacks) are launched simultaneously. The following section presents the result of the experiments.

5.1.13 DoS Attacks Results for Fingerprint Readers

Few seconds after launching the 4 DoS attacks, the recognition performances of all tested fingerprint readers deteriorated significantly. Table 5.1 summarizes the experiments results for each fingerprint reader.

Table 5.1: DoS attacks results for fingerprint readers

Models: Fingerprint Reader	Effect of DoS attacks on recognition performance of fingerprint readers
NitGen Fingerprint reader NAC 3000 (http://www.nitgen.com)	Recognition status is unstable: <ul style="list-style-type: none"> • The reader recognition response is very slow or there is no response. • The readers often disconnect from the network.
F7 Standalone Biometric Access Control Terminal (http://www.fslocks.com/f7stbiaccote.html)	
MX600 Fingerprint Access Control (http://www.miaxis.net/1070012/1/products_details.htm)	

For example, in figure 5.9; it shows that before launching the DoS attacks, the response times were less than 0.4 M/S when pinging the NitGen Fingerprint reader NAC 3000 [66]. However, the response times increased considerably and reached more than 20 ms just after launching the attacks. This is due to the fact that after

launching the DoS attacks, the reader became very busy with treating the flood of packets and consequently became unable to process the Ping requests on time

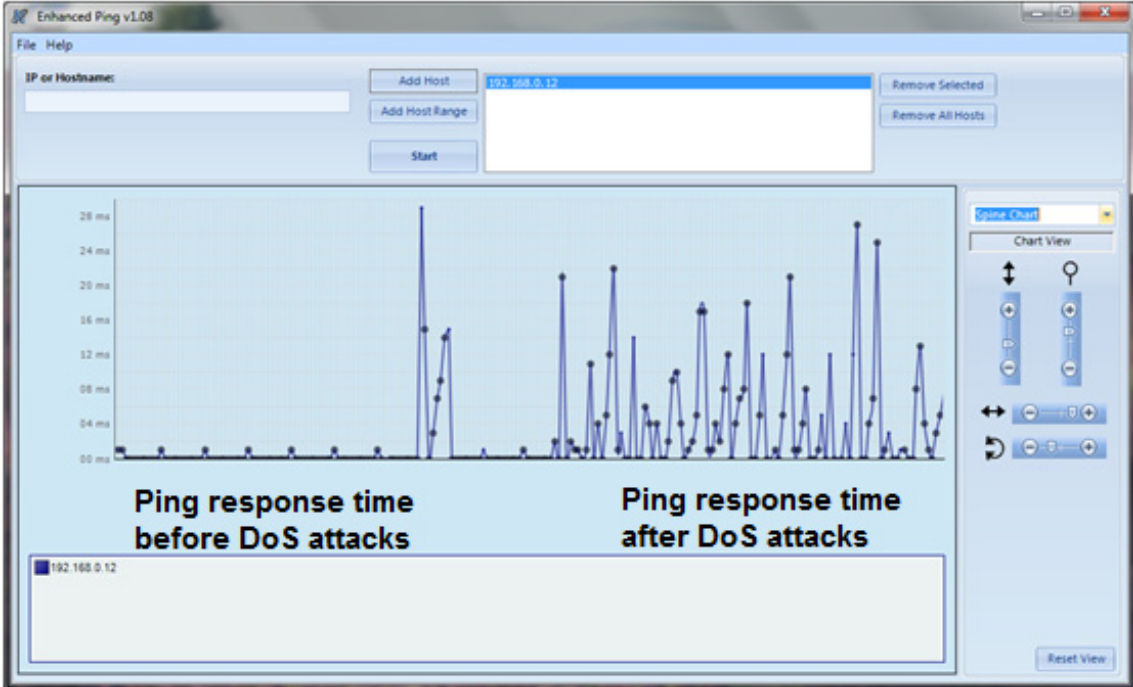


Fig. 5.9: Response time of Nitgen Fingerprint NAC 3000 before & after the DoS attacks

5.1.14 DoS Attacks Results for Iris Readers

In this experiment, the iRIS scanners are the targets of the attack hosts. The same four DoS attacks are used in this experiment. Few seconds after launching the DoS attacks, the recognition performances of all tested iRIS readers deteriorated significantly. Table 5.2 summarizes the experiments results for each iRIS reader.

Table 5.2: DoS attacks results for iris readers

Models: iRIS Scanner	Effect of DoS attacks on recognition performance of the iRIS readers
Panasonic Iris reader BM-ET330 (ftp://ftp.panasonic.com/pub/Panasonic/cctv/SpecSheets/BM-ET330.pdf)	Recognition status is unstable: -The reader recognition response is very slow or there is no response. -The readers disconnected from the network. But, when the DoS attack stopped, the readers reconnected to the network.
LG’s IrisAccess 4000 (http://www.irisid.com)	
IG-AD100® Iris Camera System, (http://www.irisguard.com)	

For example, Figure 5.10: shows that before launching the DoS attacks, the response times were less than 0.1 M/S when “Pinging” the Panasonic Iris reader BM-ET330 [67]. However, just after launching the attacks, the reader crashed and consequently there were no ping responses. The reader became unable to recognize users and completely disconnected from the network. When the DoS attack has stopped, the reader had reconnected to the network..!

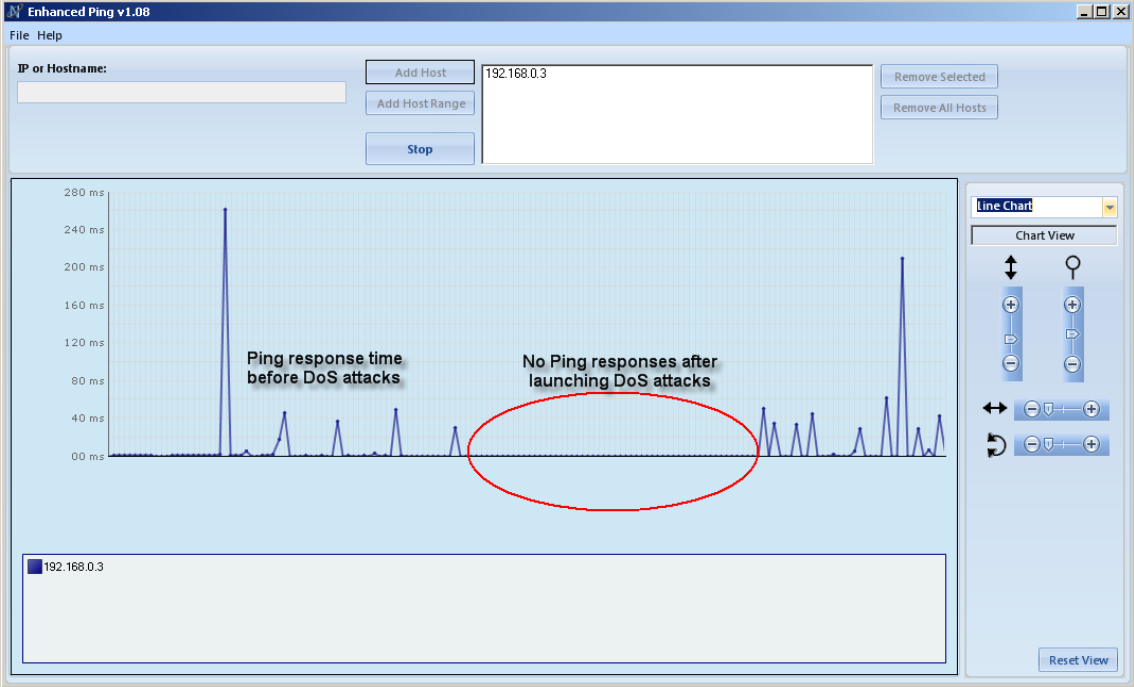
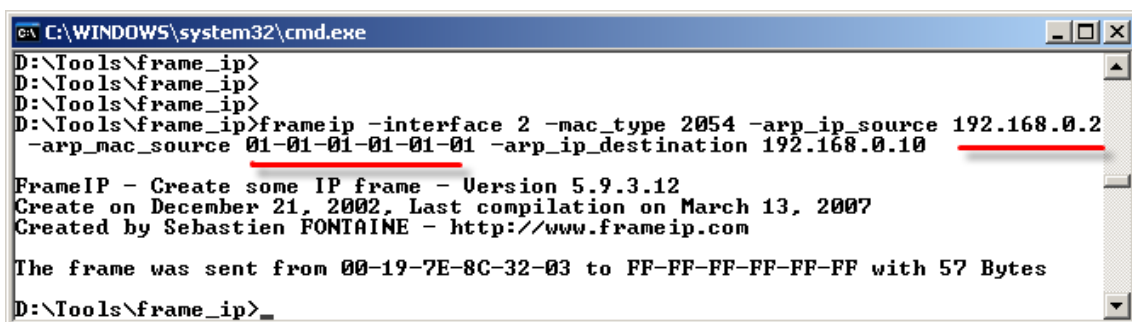


Fig. 5.10: Response time of Panasonic Iris reader BM-ET330 before & after the DoS

5.1.15 ARP cache poisoning attack results for fingerprint and iris readers

This attack consists of corrupting the ARP caches of the Biometric Readers. Network hosts with corrupted ARP caches may not be able to communicate properly with other network hosts.

We use FramelIP packet generator tool to build fake ARP packets. The packets are used to corrupt the ARP caches of the fingerprint and iRIS readers. Figure 5.11 shows the command line used to generate the fake ARP packets.



```
C:\WINDOWS\system32\cmd.exe
D:\Tools\frame_ip>
D:\Tools\frame_ip>
D:\Tools\frame_ip>
D:\Tools\frame_ip>frameip -interface 2 -mac_type 2054 -arp_ip_source 192.168.0.2
-arp_mac_source 01-01-01-01-01-01 -arp_ip_destination 192.168.0.10

FrameIP - Create some IP frame - Version 5.9.3.12
Create on December 21, 2002, Last compilation on March 13, 2007
Created by Sebastien FONTAINE - http://www.frameip.com

The frame was sent from 00-19-7E-8C-32-03 to FF-FF-FF-FF-FF-FF with 57 Bytes
D:\Tools\frame_ip>_
```

Fig. 5.11: FramelIP online command used to perform ARP cache poisoning attack

The experiments result indicates that the ARP cache poisoning attack has no effect on the tested readers. Therefore, the readers are protected from this type of attack. This is because of the simple implementation of the ARP protocol in these readers. In fact, the readers do not allow updating their ARP caches. They use static ARP cache entries, so that the entries cannot be updated by fake ARP request and replies. The ARP cache entries are created when the readers connect to the network. Once they get the MAC addresses of the biometric servers, they create static entries (IP/MAC addresses) in their ARP caches.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

In our research, we extended the study done in[54]. Where, Authors proposed an optimal algorithm to enhance the ability of the security devices or Anti-Sniffing tools against *ARP Spoofing* or what also known as *ARP Cache Poisoning*. They also place a six rules policy for the minimum requirements for such algorithm to work efficiently towards such attacks. However that algorithm was never been implemented or tested on real system.

In our research we extended the algorithm rules into machine language and implemented it in C++ language, where it was also tested in a simulation atmosphere to an actual ARP cache poisoning attacks taking place on daily basis in multiple sites over the Intranet and the Internet.

Our experiments showed that the extended algorithm could success in technically detect the 10 known and critical types of ARP Spoofing attacks and drop the harmful traffic inbound the network or the internet. on the other hand; an unexpected coincidence from the applied algorithm in the security LAB affirmed that the written algorithm after testing does NOT have a significant influence over the network speed or performance, since the fact that ARP packets have no data encapsulated in its packet and are too small in size compared to TCP/IP packets in the normal data traffic.

Furthermore, the second part was based on applying some malicious attacks such as; MiM attack[2] (Man-in-the-Middle Attack) and DoS (Denial-of-Service) against the most famous used security devices nowadays; The Biometrics devices. These devices for instance; The Fingerprint readers and iRIS (Eyes Scanner) are commonly used for authenticating the authorizing users throughout security doors or restricted access areas, such as; Military Base, Army, or critical places for example; Airports and hospitals...etc. the experiments we conducted on several models of these famous brands as security devices, showed how MiM attacks & DoS attacks are risky and how much the Biometric devices are venerable..!

It shows also that, the recognition performances of the Biometrics Scanners are significantly affected after launching the attacks. On the other hand, most of the Biometric devices we tested in the LAB were NOT significantly affected after launching the ARP Cache Poisoning attacks. Since, they are using a simple implementation of the ARP protocol. They also use static ARP caches entries; instead of using dynamic entries as it is the case in ordinary computers.

Therefore, the Biometric Devices are easy targets for malicious users. These biometric devices can be crashed or disconnected from the network by common DoS attacks. Consequently, their availability and efficiency may become questionable within any institution, and it will be difficult to rely on such devices to implement security policies.

Our research-work in this study also shows that the Biometrics devices are not designed to include basic security solutions mainly firewalls to filter the network traffic and IDS/IPS systems to detect and prevent network attacks and malicious network activities.

I believe according to the results we demonstrated in this research, there is a need to perform a whole review and standpoint on the protection of the Biometrics against such vital security flaws.

We also recommend carrying on further study on this research area that may include but are not limited to the following:

- Testing the extended algorithm that was implemented and tested in this study in a Wide Area Network [WAN] with thousand numbers of virtual hosts connected to a network backbone and high volume of ARP spoofed attacks over the switch or firewall where the extended and implanted algorithm is running, in order to mature its performance.
- Calculating the performance of the network while the algorithm is up-and-running, and cross-checking the result with our observations in the LAB and analyzing the algorithm performance on difference set up, conditions and configurations.

- It is also recommended for all security solution providers to upgrade or update their devices Operating Systems [OS] to integrate the algorithm we proposed in our study, in order to ensure that all known types of ARP Spoofing attacks are addressed and detected by the security solution. And last but not least;
- Reviewing the security protection of the Biometrics devices to ensure its credibility, integrity and reliability against the most famous and spread out thread nowadays such as ARP Spoofing, Sniffing attacks ...etc. given the fact, our study affirmed that those devices, such as; Fingerprint or iRIS, are questionable towards their vulnerability and are in doubt, and hence, the Biometrics Manufactures are accountable to assess their products and secure them against such vital attacks prior to launching the product-line into the global markets.
- As a future work, we are working on investigating the confidentiality of the data exchanged between the biometric readers and servers using the sniffing attack.

Research Publishing

- This work was presented in:
 - “The 2009 International Conference on the Current Trends in Information Technology [CTIT’09], on 16-Dec, 2009, www.ctit.ae/
 - The paper entitled: “Towards more sophisticated ARP Spoofing Detection & Prevention systems in LAN network”

- Also published on IEEE Xplore IEEE
 - Catalog Number: CFP0904J-PRT
 - ISBN: 978-1-4244-5755-7

- Advanced research will continue:
 - An abstract is being submitted and updated paper is accepted to be presenting in:
“The 11th ARC at UAE-University on September, 2010” and will be published in the conference proceeding,
http://sra.uaeu.ac.ae/Conference_11/index.htm

- A submitted paper was accepted:
 - 2nd International ICST Conference on Digital Forensics & Cyber Crime [ICDF2C], on October 4-6, 2010. Publisher: SpringerLink
 - Website: <http://d-forensics.org/>

- **APPENDIXES**

Appendix [1]: The Internet Protocol Suite; based on layers (RFC 1122)

The Internet Protocol Suite (TCP/IP protocols)

Application Layer list:

BGP - DHCP - DNS - FTP - GTP - HTTP - IMAP
- IRC - Megaco - MGCP - NNTP - NTP - POP -
RIP - RPC - RTP - RTSP - SDP - SIP - SMTP -
SNMP - SOAP - SSH - Telnet - TLS/SSL -
XMPP...etc.

Transport Layer list:

TCP - UDP - DCCP - SCTP - RSVP - ECN...etc.

Internet Layer list:

IP (IPv4, IPv6) - ICMP - ICMPv6 - IGMP -
IPsec...etc.

Link Layer list:

ARP - RARP - NDP - OSPF - Tunnels (L2TP) -
PPP - Media Access Control (Ethernet, MPLS, DSL,
ISDN, FDDI) - Device Drivers...etc.

Appendix [2]: The help menu for FrameIP Tool

FrameIP - Create some IP frame - Version 5.10.3.13

Create on December 21, 2002, Last compilation on June 02, 2009

Created by Sebastien FONTAINE - <http://www.frameip.com>

GENERAL OPTIONS

-? This help
-wait Wait after frame Default: 1000 ms
-loops Number of loops Default: 1 (0=>no stop)
-send_mode 0=Socket 1=Libpcap Default: 1
-view Show the answers Default: 1

FREE INTERFACES

0 - Generic dialup adapter
1 - Intel(R) PRO/Wireless 3945BG Network Connection (Microsoft's Packet Scheduler)
2 - Broadcom 440x 10/100 Integrated Controller (Microsoft's Packet Scheduler)
3 - Bluetooth PAN Driver (Microsoft's Packet Scheduler)
-interface Interface choice Default: 0

ETHERNET HEADER OPTIONS (-send_mode 1)

-mac_source @ Ethernet Default: a (r=>random a=>automatic)
-mac_destination @ Ethernet Default: FF-FF-FF-FF-FF-FF (r=>random a=>automatic)
-mac_type Between 0 & 65535 Default: 2048

ARP HEADER OPTIONS (-mac_type 2054)

-arp_type_hardware Header format Default: 256
-arp_type_protocol Protocol type Default: 8
-arp_length_hardware Header format Default: 6
-arp_length_protocol Protocol type Default: 4
-arp_opcode Operation type Default: 256 (1=>Request)
-arp_mac_source @ Mac source Default: 00-00-00-00-00-00
-arp_mac_source_auto Between 0 or 1 Default: 1 (1=>MAC from interface)
-arp_ip_source @ Ip source Default: 10.33.33.24 (r=>random)
-arp_mac_destination @ Mac destination Default: 00-00-00-00-00-00
-arp_ip_destination @ Ip destination Default: 192.168.101.254 (r=>random)

IP HEADER OPTIONS (-mac_type 2048)		
-ip_version	Between 0 & 15	Default: 4
-ip_ihl	Between 0 & 15	Default: 5
-ip_tos	Between 0 & 255	Default: 0
-ip_length	Between 0 & 65535	Default: a (a=>automatic)
-ip_id	Between 0 & 65535	Default: r (r=>random)
-ip_flag_zero	Between 0 or 1	Default: 0
-ip_flag_mf	Between 0 or 1	Default: 0
-ip_flag_df	Between 0 or 1	Default: 0
-ip_offset	Between 0 & 8191	Default: 0
-ip_ttl	Between 0 & 255	Default: 128
-ip_type	Between 0 & 255	Default: 1 (r=>random)
-ip_checksum	Between 0 & 65535	Default: a (a=>automatic)
-ip_source	@ Ip or host name	Default: 10.33.33.24 (r=>random)
-ip_destination	@ Ip or host name	Default: 192.168.101.254 (r=>random)
ICMP HEADER OPTIONS (-ip_type 1)		
-icmp_type	Between 0 & 255	Default: 8
-icmp_code	Between 0 & 255	Default: 0
-icmp_checksum	Between 0 & 65535	Default: a (a=>automatic)
-icmp_id	Between 0 & 65535	Default: r (r=>random)
-icmp_sequence	Between 0 & 65535	Default: r (r=>random)
IGMP HEADER OPTIONS (-ip_type 2)		
-igmp_version	Between 0 & 15	Default: 1
-igmp_type	Between 0 & 15	Default: 1
-igmp_reserve	Between 0 & 255	Default: 0
-igmp_checksum	Between 0 & 65535	Default: a (a=>automatic)
-igmp_destination	@ Ip or host name	Default: 224.0.0.1 (r=>random)
TCP HEADER OPTIONS (-ip_type 6)		
-tcp_port_source	Between 0 & 65535	Default: r (r=>random)
-tcp_port_destination	Between 0 & 65535	Default: 80 (r=>random)
-tcp_sequence	Between 0 & 2E16	Default: r (r=>random)
-tcp_acknowledge	Between 0 & 2E16	Default: 0
-tcp_offset	Between 0 & 15	Default: 5
-tcp_reserved	Between 0 & 63	Default: 0
-tcp_flag_urg	Between 0 or 1	Default: 0
-tcp_flag_ack	Between 0 or 1	Default: 0
-tcp_flag_psh	Between 0 or 1	Default: 0

-tcp_flag_rst	Between 0 or 1	Default: 0
-tcp_flag_syn	Between 0 or 1	Default: 1
-tcp_flag_fin	Between 0 or 1	Default: 0
-tcp_window	Between 0 & 65535	Default: 0
-tcp_checksum	Between 0 & 65535	Default: a (a=>automatic)
-tcp_pointeur	Between 0 & 65535	Default: 0

UDP HEADER OPTIONS (-ip_type 17)

-udp_port_source	Between 0 & 65535	Default: r (r=>random)
-udp_port_destination	Between 0 & 65535	Default: 53 (r=>random)
-udp_length	Between 0 & 65535	Default: a (a=>automatic)
-udp_checksum	Between 0 & 65535	Default: a (a=>automatic)

RIP HEADER OPTIONS (-udp_port_destination 520)

-rip_status	1:request 2:answer	Default: 2
-rip_version	Between 0 & 255	Default: 2
-rip_domain	Between 0 & 65535	Default: 0
-rip_protocol	Between 0 & 65535	Default: 2
-rip_tag	Between 0 & 65535	Default: 0
-rip_route	@ Ip	Default: 192.168.0.0 (r=>random)
-rip_mask	Mask	Default: 0.0.0.0 (r=>random)
-rip_gateway	Netx Hop	Default: 0.0.0.0 (r=>random)
-rip_metric	Between 0 & 65535	Default: 1
-rip_file_name	List of routes	Default:

OPTIONS OF THE DATA LAYER

-data_size	data size	Default: 15
-data_ascii	specify a string	Default: www.frameip.com
-data_hexa	specify some hexa	Default: 7777772e6672616d6569702e636f6d

REFERENCES

- [1] CERT: <http://www.cert.org>
- [2] Wikipedia: http://en.wikipedia.org/wiki/Man_in_the_middle_attack
- [3] WebOpedia: <http://www.webopedia.com/TERM/s/sniffer.html>
- [4] Plummer D. An Ethernet address resolution protocol, RFC 826, MIT-LCS, November1982.
- [5] Trabelsi Z., 2009. Introduction: TCP/IP Protocol & IP Address. PowerPoint lecture presented in CIT College on UAE University campus
- [6] Cisco Systems, 2009. Internetworking Technologies Handbook, Fiber Distributed Data Interface, Chapter 8, pp 125-136.
- [7] Wendell Odam, 2000. Cisco CCNA Certification Guide. Cisco Systems, Chapter 5, pp 225-228.
- [8] Trabelsi Z., 2009. ARP-RARP Protocols. PowerPoint lecture presented in CIT College on UAE University campus.
- [9] Whalen S., 2009. An Introduction to ARP Spoofing, <http://www.node99.org/papers/arpspoof.pdf>
- [10] Wikipedia, *Address Resolution Protocol*, (as of July 17, 2009, 12:47 GMT).
http://en.wikipedia.org/wiki/Address_Resolution_Protocol
- [11] Abad C., and Bonilla R., 2007. An Analysis on the Schemes for Detecting and Preventing; ARP Cache Poisoning Attacks, Proceedings of 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07), Toronto, Ontario, Canada. IEEE Computer Society 2007, pp 25-29.
- [12] Houghton M. H., 2009. The American Heritage Dictionary of the English Language, 4th Edition.
- [13] The 6th Annual Global Security Survey, 2009. Deloitte Touche Tohmatsu, London.
http://www.deloitte.com/dtt/cda/doc/content/dtt_fsi_GlobalSecuritySurvey_0901.pdf
- [14] Symbiotic Media Consortium: <http://www.symbiotic.co.ke/2009/08/it-security-is-your-data-safe-from-both-external-and-internal-attack/>
- [15] Wasiaq M., ARP Cache Poisoning Man-In-The-Middle-Attack, (as of August 16, 2009, 06:45 GMT).
http://www.ccse.kfupm.edu.sa/pages/services/pc_security/ARPCachePoisoning.doc
- [16] Trabelsi Z., and Shuaib K., 2008. A Novel Man-in-the-Middle Intrusion Detection Scheme for Switched LANs, The International Journal of Computers and Applications, ACTA Press, USA, Vol.30, 202-2195.
- [17] Trabelsi Z., 2009. ARP Cache Poisoning based Attacks. PowerPoint lecture presented in CIT College on UAE University campus
- [18] Wikipedia, *Man-in-the-Middle attack*, (as of July 17, 2009, 12:47 GMT)
http://en.wikipedia.org/wiki/Man-in-the-middle_attack

- [19] Roy D., Moazzami K., and Singh R., 2007. ARP Spoofing and Man in the Middle attack using Ettercap, School of Computer Science, University of Windsor, Canada.
- [20] Koc C., Yerubandi S., and Wanalertlak W., 2002 SSH1 MAN IN THE MIDDLE ATTACK: Computer Network Security, Oregon State University, USA.
- [21] Nelson T., 2005. Control Systems Security Center, Common Control System Vulnerability, Idaho National Laboratory.
- [22] Trabelsi Z., and Shuaib K., 2008. Spoofed ARP Packets Detection in Switched LAN Networks. In Joaquim, F., Mohammad, O. (Eds.), E-Business and Telecommunication Networks. Portugal: Springer Berlin Heidelberg, pp. 81-91.
- [23] LBNL's Network Research Group, Arpwatch: Ethernet Monitor Program, <http://www-nrg.ee.lbl.gov>
- [24] Snort: <http://www.snort.org/>
- [25] Jonathan Wilkins: <http://www.bitland.net/taranis>
- [26] (IDS) inspects all inbound and outbound network activity and identifies suspicious attack.
- [27] (IPS) provides policies and rules for network traffic along with IDS for alerting system Admin to a suspicious traffic.
- [28] Bruschi D., Ornaghi A., and Rosti E. 2003. S-ARP: a secure address resolution protocol, Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003), Las Vegas, NV, USA, pp 66 – 74, 8-12.
- [29] Omkant P., 2002 “O-ARP: a secure and fast Address Resolution Protocol”, <http://www.itbhu.ac.in/departments/comp/crypto/o-arp.pdf>
- [30] Gouda M., and Huang CT., 2003. A Secure Address Resolution Protocol, The International Journal of Computer and Telecommunications Networking, Computer Networks, Elsevier, Volume 41, Issue 1, pp 57-71
- [31] Lootah W., Enck W., and McDaniel P., 2005. TARP: Ticket-based Address Resolution Protocol, 21st Annual Computer Security Applications Conference (ACSAC 2005), Tucson, Arizona, USA.
- [32] Vipul G., Rohit T., Colin B., and González Juan M., 2005. An efficient solution to the ARP cache poisoning problem, Lecture Notes in Computer Science, Australasian conference on information security and privacy (ACISP), Brisbane, AUSTRALIA, No.10, vol. 3574, pp. 40-51.
- [33] Etienne J., 2000. ARPsec, an ARP security extension, Linux Symposium, July 19-22nd, Ottawa, Canada
- [34] Kent S., and Atkinson R., 1998. Security Architecture for the Internet Protocol, RFC 2401
- [35] Oppliger R., Hausser R., and Basin D., 2006. SSL/TLS session-aware user authentication – Or how to effectively thwart the man-in-the-middle, Computer Communications, Elsevier, Article in Press
- [36] Rivest L., and Shamir A., 1984. How to expose an Eavesdropper, Communications of the ACM 27 (4), pp 393–395

- [37] Bellovin M., and Merritt M., 1994. An attack on the interlock protocol when used for authentication, IEEE Transactions on Information Theory 40 (1).
- [38] Jakobsson M., and Myers S., 2005. Stealth attacks and delayed password disclosure. <http://www.informatics.indiana.edu/markus/stealth-attacks.htm/>
- [39] Kaliski B., and Nystrom M., 2004. Authentication: risk vs. readiness, challenges and solutions, in: Presentation held at the BITS Protecting the Core Forum. <http://www.rsa.com/rsalabs/staff/bios/bkaliski/publications/other/kaliski-authentication-risk-readiness-bits-2004.ppt>
- [40] Asokan N., Niemi V., Nyberg K., 2003. Man-in-the-middle in tunneled authentication protocols, in: Proceedings of the International Workshop on Security Protocols, pp 15–24.
- [41] Anatomy of an ARP poisoning attack. <http://www.watchguard.com/infocenter/editorial/135324.asp>.
- [42] Cisco Systems. Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, <http://www.cisco.com>
- [43] Al-Hemairy M., Trabelsi Z., and Amin S., 2009. Towards more Sophisticated ARP Spoofing Detection/Prevention Systems in LAN Networks, Proceedings of “The 2009 International Conference on the Current Trends in Information Technology (CTIT’09)”, HCT Dubai, UAE. IEEE Xplore Catalog Number: CFP0904J-PRT. ISBN: 978-1-4244-5755-7
- [44] Cisco Catalyst 3560 Series Switches, <http://www.cisco.com>
- [45] Juniper Switches EX3200 Series, <http://www.juniper.net>
- [46] Abad C., and Bonilla R., 2007. An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks, Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW’07).
- [47] FramelIP: www.FramelIP.com
- [48] WinPcap: www.WinPcap.org
- [49] Commview: <http://www.tamos.com/products/commview/>
- [50] Brown M., 2007. Guide to IP Layer Network Administration with Linux: Address Resolution Protocol (ARP), chapter 2. <http://linux-ip.net/html/ether-arp.html>
- [51] Baccala B., 1997. Connected: An Internet Encyclopedia: ARP Protocol Overview. <http://www.freesoft.org/CIE/Topics/61.htm>
- [52] Trabelsi Z., and Rahmani H., 2005. An Anti-Sniffer Based on ARP Cache Poisoning Attack. Information Security Journal, A Global Perspective, Volume 13, Issue 1, pp 23-36
- [53] Trabelsi Z., and El-Hajj W., 2007. Using a Fuzzy Logic Controller to Thwart Data Link Layer Attacks in Ethernet Networks, Wireless Communications and Networking Conference, Kowloon, Hong Kong, pp 2547-2552
- [54] Trabelsi Z., and El-Hajj W., 2009. ARP Spoofing: A Comparative Study for Education Purposes, Information Security Curriculum Development Conference (InfoSecCD’09), Kennesaw, Georgia.

[55] Woodward J., Jr., Horn C., Gatune J., and Thomas A. 2003. Biometrics; A Look at Facial Recognition, RAND Public Safety and Justice for the Virginia State Crime Commission, Richmond Virginia, USA. ISBN: 0-8330-3302-6.

[56] The MIT Technology Review in the Emerging Technologies: That Will Change the World, Ten emerging technologies that will change the world, January/February 2001 issue, <http://www.techreview.com>

[57] http://www.biometricnewsportal.com/biometrics_definition.asp

[58] Vacca, J., 2007. Biometric Technologies and Verification Systems, Butterworth-Heinemann Publisher, ISBN-10: 0750679670.

[59] Wayman, J., Jain, A., Maltoni, D., Maio, D., 2004. Biometric Systems: Technology Design and Performance Evaluation, Springer Publisher, ISBN-10: 1852335963.

[60] Chirillo, J., Blaul, S., 2003. Implementing Biometric Security, Wiley Publisher, ISBN-10: 0764525026.

[61] Mansfield T., 2001. Biometric authentication in the real world, Centre for Mathematics and Scientific Computing, National Physical Laboratory, UK (Online). Available from: http://www.npl.co.uk/upload/pdf/biometrics_psrevho.pdf

[62] Al-Raisi, A., Al-Khoury, A., 2008. Iris Recognition and the Challenge of Homeland and Border Control Security in UAE. Journal of Telematics and Informatics. 25, 117--132.

[63] Duagman, J., 2004. How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology. 14, 21--30.

[64] Advanced Networking Management Lab (ANML), Distributed Denial of Service Attacks(D-DoS) Resources. Available from: <http://anml.iu.edu/ddos/types.html>

[65] SYN-Flood: http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm

[66] Nitgen Fingerprint reader NAC 3000, Specification Sheet, <http://www.nitgen.com>

[67] Iris reader BM-ET330, Specification Sheet, <ftp://ftp.panasonic.com/pub/Panasonic/cctv/SpecSheets/BM-ET330.pdf>