An ARP - ICMP probe packet based scheme to prevent ARP Poisoning and IP Exhaustion attacks

A dissertation submitted

to

School of Computer and Information Sciences in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

Nikhil Tripathi
(12MCMB10)

University of Hyderabad,
Hyderabad - 500046
INDIA

Certificate

This is to certify that the dissertation entitled **"An ARP - ICMP probe packet based scheme to prevent ARP Poisoning and IP Exhaustion attacks"** submitted by **NIKHIL TRIPATHI** bearing Reg. No. 12MCMB10, in partial fulfillment of the requirements for the award of Master of Technology in Information Technology is a bonafide work carried out by him under my supervision and guidance. The dissertation has not been submitted previously in part or in full to this or any other university or institution for the award of any degree or diploma.

Dr. B. M. Mehtre
Project Supervisor,
IDRBT, Hyderabad

Dean,
School of Computer and Information Sciences,
University of Hyderabad

DECLARATION

I, Nikhil Tripathi, declare that this dissertation entitled **An ARP - ICMP probe packet based scheme to prevent ARP Poisoning and IP Exhaustion attacks**, submitted by me under the guidance and supervision of Dr. B. M. Mehtre, Associate Professor, IDRBT, is a bonafide work. I also declare that it has not been submitted previously in part or in full to this university or other university or institute for the award of any degree or diploma.

Signature of the student
NIKHIL TRIPATHI
Reg No: 12MCMB10

Date:

ACKNOWLEDGMENTS

I would first like to thank God who gave me the grace and privilege to pursue this program and successfully complete it in spite of many challenges faced. I express my heart-felt gratitude to my respected guide, **Dr. B. M. Mehtre**, for his committed guidance, valuable suggestions, constructive criticisms and precious time that he invested throughout the work. His stimulating suggestions and encouragement helped me at all the time of my research and writing of this thesis.

I am extremely thankful to **Shri B. Sambamurthy, Director, IDRBT** for providing me infrastructural facilities to work in, without which this work would not have been possible.

My special thanks to **Prof. Arun K. Pujari, Dean, SCIS** for the keen support and consistent encouragement in our academic activities.

I would also like to thank all the referees who reviewed this work as pieces of it were submitted to various conferences and journals. Their detailed reviews, constructive criticism and excellent advice have improved both the presentation and content of this thesis.

At the outset, I would like to thank the **University of Hyderabad** for providing all the necessary resources for the successful completion of my course work. At last, but not the least I thank my classmates and other students of SCIS for their physical and moral support.

Finally, I nostalgically account for the silent sacrifice and heartening inspiration from my parents that helped me to steer through the rough weathers smoothly.

ABSTRACT

Today, insider threats are considered as one of the biggest challenges for the organizations to protect their critical infrastructure. Insider threats can be posed either by malicious insiders who intentionally want to cause harm or by trusted employees who unintentionally, due to unawareness, cause damages to the organization. Insider threats range from low level attacks like ARP Poisoning to most advanced attacks like DoS/DDoS Attacks. The low level attacks are used to launch the most advanced high end attacks. If these lower level threats are prevented in the starting phase itself, we can stop the advanced level attacks from happening.

Several solutions have been proposed in the literature to prevent such attacks. However, some solutions are effective in a special set of scenarios while others are rather suited for scenarios belonging to a different band. As new techniques of ARP poisoning have evolved with time, there is need to propose new solutions.

In this dissertation, a new scheme has been proposed to mitigate the ARP Poisoning attacks. This scheme is programmed using raw socket programming in python. The scheme uses Internet Control Message Protocol (ICMP) and ARP probe packets for the detection and prevention of ARP Poisoning and IP Exhaustion problem. The scheme does not require any modification in the existing ARP specifications. Also, the scheme can be used even when IP aliasing is configured for some of the interfaces. Being a decentralized scheme, it does not create a single point of failure problem. The experimental results are also prsented to support the proposal. Also, our proposed scheme is compared with the existing solutions to show that our proposed scheme is better and more effective than previously proposed solutions.

Keywords: Insider Threats, ARP Poisoning, Man-In-The-Middle attacks, IP Exhaustion, Cyber Defense, Network Security,

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

- **ARP**: Address Resolution Protocol
- **ICMP**: Internet Control Message Protocol
- **IP**: Internet Protocol
- **DoS**: Denial of Service
- **DDoS**: Distribute Denial of Service
- **MAC**: Media Access Control
- **RFC**: Request for Comment
- **MITM**: Man-In-The-Middle
- **AKD**: Authoritative Key Distributor
- **DHCP**: Dynamic Host Configuration Protocol
- **LTA**: Local Ticket Agent
- **ACS**: ARP Central Server
- **DES**: Discrete Event System
- **DG**: Default Gateway

CHAPTER 1
INTRODUCTION

1.1 INSIDER THREATS

An insider threat is defined by the Computer Emergency Response Team at Carnegie-Mellon University (CERT) [1] as "a malicious insider who is a current or former employee, contractor, or business partner who has or had authorized access to an organization's network, system or data, and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information system". While insider threats may be from such malicious insiders who intentionally cause harm for their personal gain or revenge, insider threat can also be posed by trusted employees who unintentionally, through negligence, cause financial or reputational damages to the organization.

The insider threat has become a significant issue. Insider threats have grown because the value of data has increased, giving insiders more incentive to steal data [2]. Since insider attacks are much more targeted because they know where the data is actually placed, these attempts are likely to inflict a greater impact compared to external threats [3]. In fact, the 2010 Verizon Data Breach Investigations Report along with other studies concluded that it is pricier to fix insider attacks compared to external threats [4]. This is also confirmed by the e-crime Survey and Ponemon Institute's 2010 Cost of Cyber Crime Study [5]. One among these attacks are DoS/DDoS Attacks [6].

There have been considerably more reported insider threat incidents over the past few years. According to the 2009 e-Crime Watch Survey[7] in which 523 organizations were involved, 51% of the organizations experienced an insider attack, which increased from 39% three years ago. Since these were only reported incidents

of attacks, it is likely more than 51% of systems experience such attacks. From the recent Cyber-Ark Global Survey [8] conducted in the spring of 2011 with 1,422 IT staff and C-level professionals, 16% of the surveyed individuals believe that insiders have stolen highly sensitive and valuable intellectual property, such as customer lists and product data, which have been reassigned or sold the organizations' competitors.

In some cases, it is also found out that the employees are paid by various competing organizations to launch different attacks on the employees' home organization's local networks so that the organization come to grinding halt. These attacks range from low level attacks like Address Resolution Protocol (ARP) Poisoning, IP spoofing to more advanced attacks like DoS/DDoS attacks. ARP Poisoning is used mainly for two purposes. One is to steal the login credentials and other is to disrupt the complete network services. So, ARP Poisoning directly affects the confidentiality and the availability.

From the above statistics, one can understand that while insider threat has always existed in organizations, it has increasingly become a substantial topic that should be better handled. Managing insider threat may be difficult as insider threats are influenced by a combination of technical, behavioral and organizational issues. This requires organizations to devise layered defense plans, consisting of policies, procedures and technical controls.

In this chapter, we have explained the working of ARP protocol and how it is exploited to launch ARP Poisoning attacks.

1.2 OVERVIEW OF ADDRESS RESOLUTION PROTOCOL (ARP)

In computer networks, every interface of a computer is assigned a physical (MAC) address and a logical (IP) address. Address Resolution Protocol (ARP) is responsible for resolving the IP addresses into corresponding MAC addresses. Request for Comment (RFC) 826 [9] defines the specification of the ARP protocol. Within a LAN, the computers use MAC addresses for the purpose of communication. For that purpose, ARP provides two different types of messages which are exchanged by the computers so as to resolve the IP addresses into corresponding MAC ad-

dresses. These messages are: *ARP Request* and *ARP Reply* messages. ARP Request messages are generally broadcasted to get MAC address for the corresponding IP address. In response, ARP Reply is sent by the computer which holds that particular IP address. When the ARP request sender receives this reply, it stores the IP-MAC binding in its primary ARP cache. This cache is a volatile memory. The IP-MAC bindings can also be entered manually also in the cache.

If host A wants to communicate with host B present in the same LAN, A will look for an entry of B's IP-MAC binding in its ARP cache. If the entry is already present, A will initiate the communication. Otherwise, as shown in Fig. 1.1a, it will broadcast an ARP Request to receive B's MAC address. In response, B will send a unicast ARP Reply to A as shown in Fig. 1.1b.

1.3 OVERVIEW OF ARP POISONING

Some of the loopholes make the ARP protocol quite vulnerable to different network attacks. These loopholes are - *unauthenticated* and *stateless* nature of the ARP protocol. Hackers easily exploit these loopholes for launching other higher level attacks. Since it is the insider who launches the attack, ARP poisoning attacks fall under the category of insider threats.

Fig. 1.2 represents the ARP poisoning attack launched by C. C is sending spoofed ARP replies to A by claiming itself as B. Also, C is sending spoofed ARP replies to B by claiming itself as A. As a result, C will get the MITM position for the traffic between A and B. Once C will get MITM position, it can launch different network attacks like DoS attacks, Session Hijacking, etc. C can launch the IP Exhaustion attack also by claiming that all IP addresses (not in use currently) within the LAN belong to his/her MAC address.

These loopholes and the resultant network threats always motivated researchers to propose different strategies so as to mitigate ARP Poisoning attacks and also, the higher level attacks. Nevertheless, attackers are still able to bypass the security furnished by these solutions. In recent scenario, it has been discovered that the attackers practice some of the latest and the obscure but most basic tricks so as to fool the above solutions. So, there should be a robust and efficient mitigation

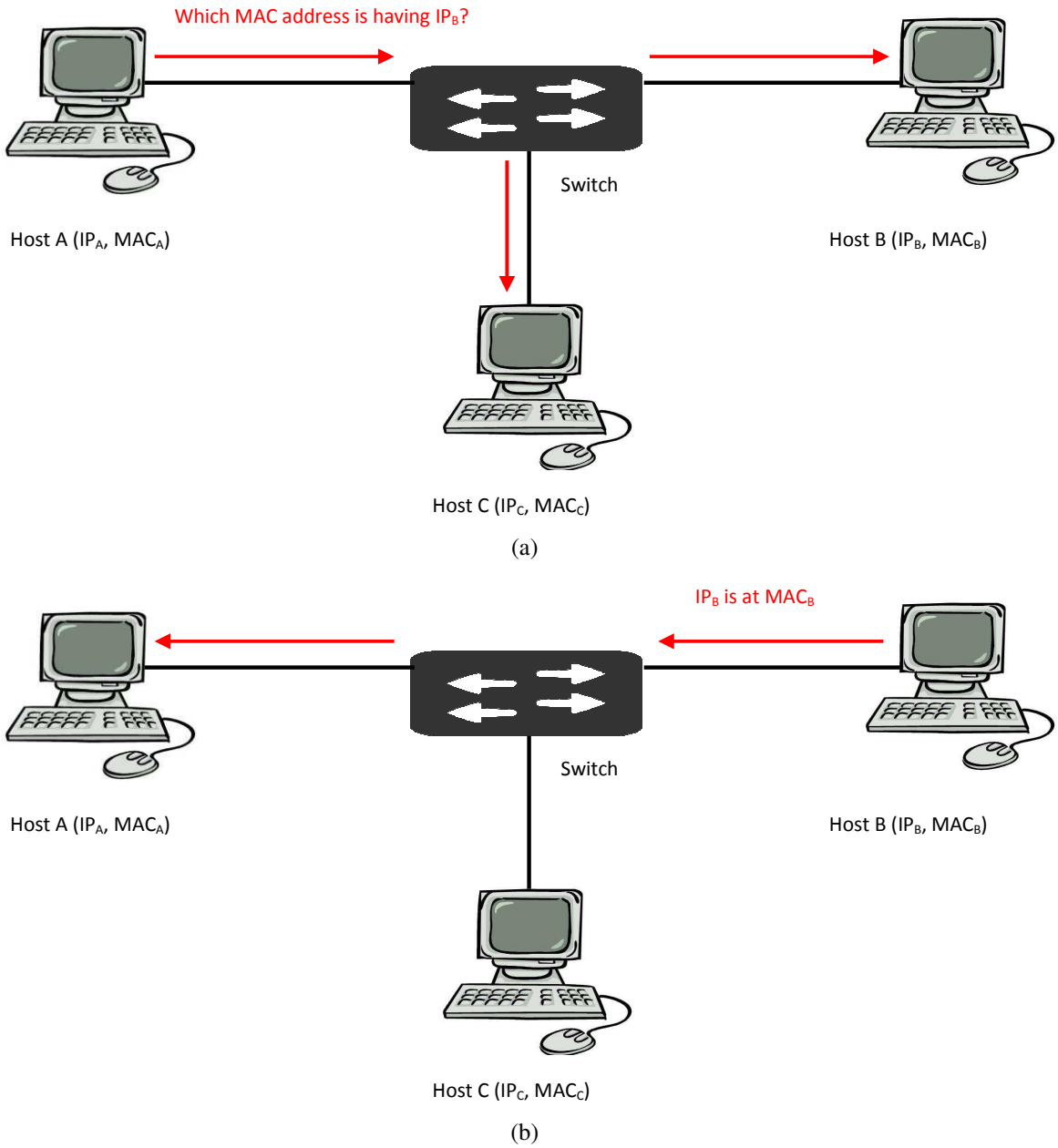


Fig. 1.1: (a) Host A broadcasts ARP Request for Host B, (b) Host B sends a unicast ARP Reply to Host A

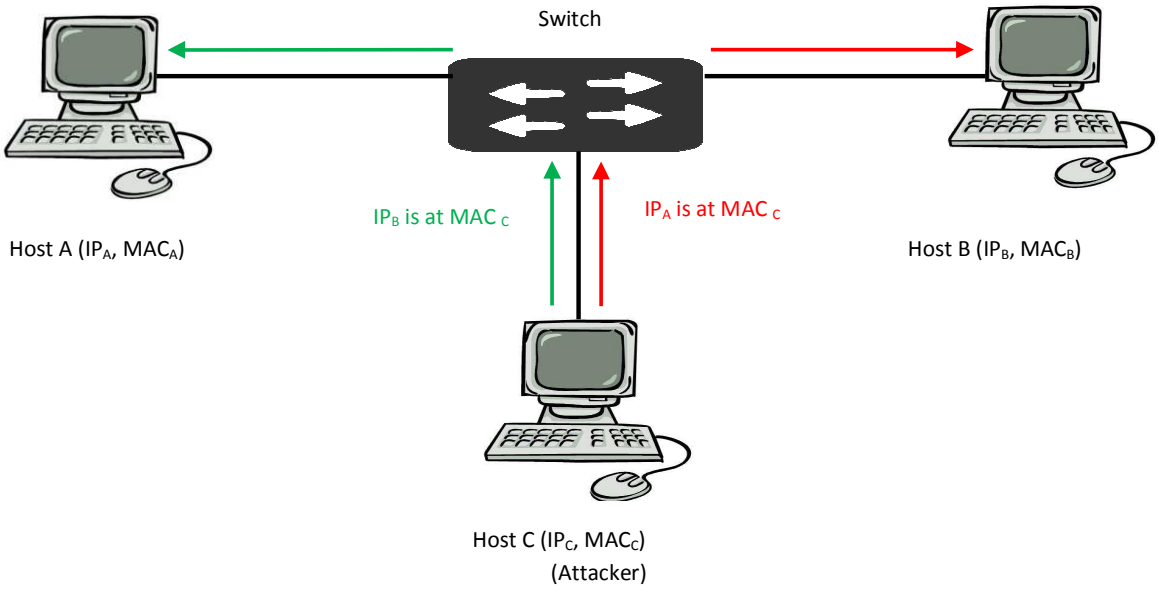


Fig. 1.2: Host C performing ARP poisoning on A and B

technique which can protect the network from these attacks at a minimal possible cost, i.e., it should neither require a change in underlying infrastructure nor should it overload the network with huge traffic.

1.4 IP EXHAUSTION ATTACKS

Some of the proposed solutions within the literature consider that if a MAC address, MAC (X), is the first one to claim that an IP address, IP (P), belongs to it, IP (P) - MAC (X) mapping is a genuine one. Thereafter, the subsequent ARP messages claiming IP (P) belong to any other MAC address will be considered as the fake one.

The IP Exhaustion problem is essentially a type of ARP poisoning attack. The attacker broadcasts multiple ARP messages on the behalf of all unused (i.e. Not alive) IP addresses in the subnet. The attacker claims that all the unused IP addresses inside the subnet belong to my MAC address. When the other hosts receive this message, they update their primary ARP cache accordingly.

Suppose that hosts inside the subnet are executing the above mentioned scheme. If an attacker launches the IP Exhaustion attack, the hosts will update their cache accordingly. Now, hosts will assume that all the possible IP addresses inside the subnet are currently in use. So whenever, a genuine host will come into the network

and it will send an ARP message, all other hosts will consider the new host as illegitimate one. Finally, a genuine host may get banned while on the other hand, the attacker can enjoy total exemption within the network.

1.5 ORGANIZATION OF THESIS

The rest of the thesis is organized as follows: Chapter 2 discusses the related works proposed in the literature. Chapter 3 deals with the theoretical explanation of various components of the proposed scheme. Chapter 4 focuses on the implementation details of the proposed scheme on the testbed along with the traffic overhead analysis. Chapter 5 deals with the comparison of the proposed scheme with the existing schemes that are rather popular in the literature. Finally, Chapter 6 provides the conclusion for the dissertation.

CHAPTER 2
LITERATURE SURVEY

In this chapter, some of the proposed schemes reported in the literature to prevent ARP poisoning are discussed below:

2.1 STATIC ARP ENTRIES AND MONITORING ARP CACHE

Several solutions have been proposed to prevent the networks from ARP poisoning attacks. One of the simplest ways is to insert static IP-MAC entries into ARP cache [10]. The static IP - MAC mappings are stored into the hosts' primary ARP cache. X. Hou et al. [11] proposed a scheme which involves monitoring of ARP cache. This scheme is also dependent on static entries into the default gateway. These schemes are quite effective if the hosts in a small local network are connected permanently without changing their IP addresses.

However, for large networks, which are quite dynamic in nature; this scheme will lead to unmanageability of the network. It is almost impossible to make static entries if the network contains thousands of the hosts.

2.2 CRYPTOGRAPHY BASED SCHEMES

Different improvements in ARP specifications are presented in schemes like Secure ARP [12], Ticket based ARP [13] and Enhanced ARP [14]. Secure ARP is based on the concept of public/private key certificates. These digital certificates are used for the authentication of every ARP replies within the network. The Authoritative Key Distributor (AKD) acts as a central server to distribute the public keys to different hosts. This scheme makes use of Secure-DHCP (Dynamic Host Configuration Protocol) instead of usual DHCP.

The drawback of these schemes is that these require changes in the implementation of ARP protocol. As a result, the schemes are not compatible with the existing networks. Also, the AKD, being centralized, may lead to single point of failure which leaves a network insecure. Ticket based ARP involves usage of a ticket with each ARP message. This ticket is distributed by a Local Ticket Agent (LTA). However, this LTA may also create a single point of failure.

2.3 KERNEL BASED PATCHES

The Antidote approach [15] is one of the most popular kernel based patch proposed to prevent ARP poisoning at an individual host level. According to Antidote scheme, when a host receives an ARP reply whose MAC address differs from the previously cached one, it tries to check if the previously learnt MAC address is still alive. If the previously learnt MAC address is still alive, the updated binding is rejected and the offending MAC address is appended to a list of banned addresses.

However, if attacker comes before the real user within the network, he can already poison the ARP cache. This causes the real MAC address to get banned.

2.4 PASSIVE DETECTION

ARPWATCH [16] is one of the most popular tools which work as a passive detection tool. It sniffs the ARP Requests/Replies on the network and constructs a MAC-IP address mapping database. If it notices a change in any of these mappings in future ARP traffic, the alarm is raised concluding that an ARP spoofing attack is going on.

The problem with the passive detection approaches occurs due to the time lag between the learning of IP-MAC bindings and attack detection. If attacker started the ARP spoofing attack before the detection tool was started for the first time, the tool will learn the spoofed IP-MAC bindings and thus, fake IP-MAC bindings will be stored in address mapping database. Also, the passive detection technique is not able to judge whether the newly seen address mapping is because of a spoofing attempt or the previously learnt one was actually a spoofed one.

Another problem with the passive detection techniques is the unreliability. If attacker sends a flood of spoofed ARP messages having some random IP-MAC

bindings, the legitimate hosts will consider them as new hosts. As a result, the hosts will store these random bindings in their address mapping database. This will result into a very big database having lots of invalid entries.

2.5 USAGE OF CENTRALIZED DETECTION AND VALIDATION SERVER

Sumit Kumar and Shashikala Tapaswi [17] proposed a centralized technique for detection and prevention of ARP poisoning. In this scheme, an ARP Central Server (ACS) validates the ARP tables' entries of all the hosts within the network. Clients also maintain a secondary long term cache in this scheme. Also, Gao Jinhua and Xia Kejian [18] proposed an ICMP protocol based detection algorithm for ARP spoofing. This algorithm collects and analyzes the ARP packets and then injects ICMP echo request packets so as to probe the malicious host according to its response packets. This algorithm is also based on a central database available at Detction host.

However, these techniques do not address the IP exhaustion problem. Moreover, these techniques are centralized in nature. Thus, the failure of ACS server may leave the network insecure.

2.6 ICMP PACKETS AS PROBE PACKETS

Using probe packets is another solution to prevent ARP poisoning attack. ICMP packets are also used as probe packets [19]. The response of these probe packets are then captured to verify the previously received ARP messages.

However, attackers have a very destructive trick that they can use to fool probe packet based techniques. Attackers can send a flood of spoofed ARP messages with maximum possible speed by using the available bandwidth of the network. As a result, the victim will not be able to receive the probe reply sent by a legitimate host. So, this approach fails once attacker launches a flood of spoofed ARP messages. Also, the attackers can disrupt the network services for a host by sending it the same flood because the primary ARP cache will be updated continuously with the spoofed fake binding.

2.7 USING HOST BASED DISCRETE EVENT SYSTEM

Ferdous A. Barbhuiya et al. [20] proposed a scheme using host based Discrete Event System (DES). The scheme is based on a DES model for the system under normal condition and also under each of the failure conditions. Along with that, a state estimator called diagnoser (or detector, if only detection of failure is required) is designed which observes events generated by the system to decide whether the states through which the system traverses correspond to the normal or faulty DES model.

However, this scheme also uses ARP Requests as probe packets. As a result, the attackers can fool this scheme by sending the flood of spoofed ARP messages. Since the victim will receive the flood of spoofed messages at maximum possible speed, it will not receive the legitimate probe response. The proposed scheme is also ineffective against IP Exhaustion attack. An attacker can easily send spoofed ARP Replies saying that different IP addresses (which are not yet verified by the victim host) belong to his/her MAC address. When victim will send probe ARP Requests for these replies to verify them, attacker can send spoofed replies again. As a result, the victim host will store these bindings into the verification table. So, the new upcoming hosts' ARP Requests/Replies will be added to the victim host's spoofed table directly though these new hosts are genuine ones.

So, the desirable characteristics of an ARP detection and prevention scheme are:

1. It should resist the flood of spoofed ARP messages so as to detect and prevent the attack efficiently.
2. It should prevent IP Exhaustion problem.
3. It should not modify the existing ARP specifications and should be backward compatible with the existing network.
4. It should not create a single point of failure problem.
5. It should be compatible with the IP aliasing configurations within the network so as to prevent false alarms.

CHAPTER 3

ARP - ICMP PROBE PACKET BASED SCHEME TO MITIGATE ARP POISONING AND IP EXHAUSTION ATTACKS - A NEW APPROACH

We had proposed an ICMP based secondary cache approach to detect and prevent ARP Poisoning [21]. It was shown how the entering and existing algorithms could detect and prevent the ARP poisoning and IP Exhaustion attacks. The proposed solution neither required any change in ARP specification nor did it create the single point of failure. It was also backward compatible to existing networks. However, the solution had various drawbacks. Some of them are:

1. *No defense against the invalid IP-MAC bindings sent by attacker:* In the previously proposed scheme, when host receives an ARP message, the existing algorithm checks the secondary table to find any entry related to the same IP or MAC address. If there is no such entry, the binding is stored blindly into the secondary ARP table without sending any probe request back to validate the received binding. So, all the random or invalid bindings sent by the attacker will be stored into the table. This may result into the unusual large size of the secondary table which increases the searching time of the MAC-IP mappings into the table.

2. *ICMP based probing to learn IP-MAC bindings:* The scheme sends ICMP echo requests to learn the IP-MAC bindings of other hosts. This probing was quite inefficient. It takes long time to learn the bindings.

3. *No defense against flood of spoofed ARP messages:* The flood of spoofed ARP messages sent by the attacker can fool the scheme.

4. *False Alarms due to IP Aliasing configuration:* If some of the MAC addresses

are configured to use IP Aliasing within the network, the proposed solution generates alarms for the IP address which are allotted to single MAC address.

5. *Multiple duplicate entries during simultaneous execution of entering and existing algorithm:* The previously proposed scheme writes duplicate entries in the secondary table multiple numbers of times during the simultaneous execution of entering and existing algorithm. Thus, the simultaneous execution was not suitable for the previous scheme. So, an attacker can launch the attack at the time when entering algorithm is executing alone because entering algorithm is responsible only for learning MAC-IP mappings.

The ARP-ICMP based secondary ARP table used in this proposed scheme has been adapted from the secondary cache described in previously proposed approach [21]. The new proposed scheme has the advantage of the previous scheme and on the other hand, does not suffer from the various drawbacks mentioned above.

In this dissertation, we proposed a decentralized scheme to detect and prevent ARP poisoning and IP Exhaustion attacks. The scheme uses ARP and ICMP protocols for the purpose of probing. The ARP probing is used during learning phase (new host entering algorithm's execution) while ICMP probing is used while validating ARP messages (existing host algorithm's execution). To make the scheme backward compatible with the existing network, the scheme will create the secondary table as a text file. The entries are made and validated in this table using ARP and ICMP protocol. The secondary table makes sure that it does not contain different multiple entries for a single IP address or for a single MAC address. Preventing multiple entries for a single IP address mitigates ARP Poisoning while preventing multiple entries for a single MAC address mitigates IP exhaustion attack. According to the entries present in secondary table, the primary cache is also updated either by adding the static entries or by deleting it. The scheme executes two different algorithms - *new host entering algorithm* and *existing host algorithm*. New host entering algorithm will be executed by a host that enters into the network when other hosts (including attacker) are already present. Existing host algorithm will be continuously executed by the hosts which are present into the network. A host which has just entered into the network will execute the new host entering algorithm

and existing host algorithm in parallel. New host entering algorithm is executed to learn the IP-MAC bindings of the live hosts within the network and store it in host's secondary ARP table. Once the IP-MAC bindings of all live hosts are learnt, the new host entering algorithm's execution will come to an end. Then the existing host algorithm will be responsible for the validation of ARP requests and replies so as to prevent the attacks. Both the new host entering algorithm and existing host algorithms also check for the flood of spoofed ARP messages. If such condition occurs, the alarm module will be activated. The alarm module is integrated with the scheme so that it can alert the network administrators for such possible attacks. Also, the IP aliasing table allows network users to define the MAC addresses on which IP aliasing is configured. Those MAC addresses will be treated differently by the algorithms so as not to raise false alarms.

We made some assumptions regarding the LAN:

1. Legitimate hosts should not drop the ICMP echo requests and response with ICMP echo replies within a specific time interval.

2. Legitimate hosts should response to ARP requests within a specific time interval.

The proposed scheme is basically a tool built using python and raw socket programming. To craft and send customized packets, we used Scapy [22] as a packet manipulation tool. This proposed scheme can be used by all hosts within the LAN. There are five basic components of the scheme:

1. *IP aliasing table:* This table allows a network user to implicitly define the MAC address and the IP addresses associated with it, in case IP aliasing is configured on that MAC address.

2. *Secondary ARP Table:* This table contains the entries of IP-MAC bindings of the live hosts. It is used by the new host entering algorithm and existing host algorithm to validate the ARP messages received by the host. The table contains only one entry for a single IP address or for a single MAC address so as to prevent ARP poisoning and IP exhaustion attack.

3. *New Host entering Algorithm:* New Host entering algorithm will be executed by the new hosts which are entering into the network. This algorithm is responsible for the storing of the IP-MAC bindings of different hosts into the secondary ARP table.

4. *Existing host Algorithm:* Existing host algorithm will be executed by all the hosts which are present into the network. New hosts will execute this algorithm in parallel with the execution of new host entering algorithm. Existing host algorithm is responsible for validating the ARP messages received by the host. It uses ICMP probe packets to validate the ARP messages. It is also responsible for the storing of IP-MAC bindings of the new hosts into the secondary ARP table.

5. *Alarm module:* If the algorithms detect the attack, the alarm module will be activated and the alarm gets raised. The alarm will alert the network administrator about the attack.

In the next subsections, these components are elaborated:

3.1 IP ALIASING TABLE

Generally, IP aliasing is configured on some special purpose hosts, e.g. Web Servers, Mail servers, etc. IP addresses of these hosts are not changed frequently in normal scenario. Network users can provide details about the interfaces' MAC addresses which have been configured to use IP Aliasing. The users can make entries for different MAC addresses and the multiple IP addresses mapped to each of them. Also, a network administrator can distribute this table to users because the IP Aliasing based bindings are not changed frequently so as to manage the network properly and efficiently.

3.2 SECONDARY ARP TABLE

Similar to IP Aliasing table, secondary ARP table is also a text file which contains the MAP-IP mappings of live hosts within the network. After receiving and validating new mappings, the algorithms store these mappings into secondary table. Then

these mappings in secondary table are used to validate the incoming ARP messages. The table contains only one entry for a single IP address or a single MAC address so as to prevent ARP poisoning and IP exhaustion attack.

3.3 NEW HOST ENTERING ALGORITHM

As discussed earlier, a new host that has just entered into the network executes the new host entering algorithm. The user of the host will implicitly define the network address and subnet mask so that the algorithm can determine all possible IP addresses within the network. The algorithm first broadcasts an ARP reply which contains the IP-MAC binding of the host itself. Using existing host algorithm, this binding will be validated and stored by all other hosts into their secondary ARP table. Then the new host entering algorithm will start sending the ARP Requests for all the possible IP addresses one by one. Thus, all the live hosts will respond back with ARP Replies containing their IP-MAC bindings. As a result, the IP-MAC bindings of all live hosts will be stored into the secondary ARP table of the new host after validation using New Host Entering algorithm/Existing Host algorithm (since both are executing in parallel). Also, the algorithm sniffs 3 packets from the traffic within 500 milliseconds. If all the packets belong to ARP messages, it concludes that a flood of ARP messages is being sent. Thus, it will raise the alarm by activating the alarm module. The new host entering algorithm's complete working is explained in the example given below.

Working Example: In this example, we are assuming that there is no ARP flood sent by the attacker. We have discussed it while considering different attack scenarios in Section 3.3.1.

Suppose X is a host that has just entered into the network. It started the new host entering algorithm's execution. The algorithm broadcasted an ARP Reply claiming X's IP address (IP_X) is associated to X's MAC address (MAC_X). All other hosts validate this reply using existing host algorithm and if found genuine, store it in secondary ARP table. Now the new host entering algorithm starts sending the ARP Requests for all the possible IP addresses one by one. The live hosts respond back with ARP Replies. Let, at time t, X is going to send an ARP Request for an IP

address IP_Y. Before sending the ARP Request, the new host entering algorithm will search for an entry of IP_Y into the IP Aliasing table. If there is such an entry into this table, the new host entering algorithm will directly jump over to next IP address without sending the ARP Request for the current IP address IP_Y. However, if there is no such entry into the aliasing table, the new host entering algorithm will send the ARP Request for IP_Y. If IP_Y is not assigned to any host, X will not receive ARP Reply back. But if IP_Y is assigned to a host (say, Y), Y will respond back with an ARP Reply. Let Y's MAC address be (MAC_Y). Now the new host entering algorithm at X will search for an entry of (MAC_Y) into the secondary ARP table and IP aliasing table both. This gives rise to two situations:

1. *No entry found for MAC_Y in secondary table:* If no entry is found for MAC_Y in secondary table, the new host entering algorithm will search for MAC_Y into IP Aliasing table. This further gives rise to two situations:

    a. *No entry found for MAC_Y in IP Aliasing table:* If there is no such entry in IP Aliasing table also, the new host entering algorithm will store the binding $MAC_Y - IP_Y$ in its secondary ARP table.

    b. *Entry found for MAC_Y in IP Aliasing table:* If such entry is found in IP aliasing table, the new host entering algorithm will activate the alarm module so as to raise the alarm. The alarm module will respond that "MAC_Y is creating IP Exhaustion problem" because user has already defined the IP addresses mapped to MAC_Y in IP aliasing table. So the algorithm will consider that the ARP Reply having new mapping of MAC_Y is sent by the attacker.

2. *Entry found for MAC_Y in secondary table:* An entry for an IP or MAC address is made by new host entering algorithm/Existing host algorithm in secondary table iff it will not be present in IP aliasing table. So we need not check IP aliasing table because the entry is present in secondary table.

    If an entry is found for MAC_Y in secondary table, the new host entering algorithm will look for the IP address mapped (say, IP_{YB}) to it. If IP_{YB} and

IP_Y are same, the ARP Reply will simply be ignored and the new host entering algorithm will directly jump over to next IP address and send the ARP Request for it. Since the existing host algorithm is also executing in parallel with the new host entering algorithm, it may happen that existing host algorithm has already made an entry for the mapping $MAC_Y - IP_{YB}$. That is the reason we matched these two IP addresses (IP_{YB} and IP_Y) to prevent duplicate entries multiple times. However, if IP_{YB} and IP_Y are different, the new host entering algorithm will start the probing process. The new host entering algorithm will send a probe ARP Request to the older mapping (i.e. $MAC_Y - IP_{YB}$) present in secondary table. If the algorithm receives back the ARP Reply, it means that the older mapping is still valid i.e. MAC_Y is possessing IP_{YB}. Thus the new mapping i.e. $MAC_Y - IP_Y$ is a fake one. As a result, the new host entering algorithm will activate the alarm module. The alarm module will respond that "MAC_Y is creating IP Exhaustion problem". This is because the older mapping is still valid and it is impossible for one MAC address to possess two IP addresses at the same time without IP Aliasing being configured for that MAC address (no entry for MAC_Y in aliasing table). However, if the algorithm does not receive the ARP Reply back, it means that the older mapping exists no longer. Thus, the new host entering algorithm will simply replace IP_{YB} with IP_Y in the secondary table and move to next IP address for sending ARP Request.

The following short notations are used in the algorithms:

SEC_{table} - Secondary ARP Table; $ALIAS_{table}$ - IP Aliasing Table; IP_{FIRST} - First IP address of the subnet (excluding network address); IP_{LAST} - Last IP address of the subnet (excluding broadcast address); IP_{POSS} - Possible range of IP addresses within the subnet; $MAC_{FLOODER}$ - MAC address of the host launching ARP flood; $IP_{FLOODER}$ - IP address for which flood is being launched; $ALARM_{MOD}$ - Alarm Module; IP_{CURR} - Current IP address which is being considered; MAC_{CURR} - MAC address which is possessing IP_{CURR}; IP_{BCURR} - IP Address mapped to MAC_{CURR} in SEC_{table}; ARP_{REQ} - Probe ARP Request; ARP_{REP} - Probe ARP Reply; IP_X - X's IP address; MAC_X - X's MAC address;

Formal Discussion of New host entering Algorithm executing at host X is given below:

Algorithm 1: **New host entering Algorithm**

Input (as arguments): Network Address and subnet mask

Output: If validation passed, make MAC - IP entries into the SEC_{table}. Otherwise, activate $ALARM_{MOD}$.

1: Send one ARP_{REP} claiming $MAC_X - IP_X$.

2: Calculate IP_{POSS} for the subnet. Initialize IP_{CURR} = IP_{FIRST}.

3: while (IP_{CURR}! = IP_{LAST})

4: Search for an entry of IP_{CURR} in $ALIAS_{table}$. If entry found, go to Step 17

5: Sniff 3 packets from the network within 0.5 seconds. If all 3 packets are not ARP packets, go to Step 7.

6: Extract $MAC_{FLOODER}$ and $IP_{FLOODER}$. Activate $ALARM_{MOD}$ by sending the message "ARP flood detected by $MAC_{FLOODER}$ for $IP_{FLOODER}$"

7: Send ARP_{REQ} for IP_{CURR}.

8: If corresponding ARP_{REP} received, extract MAC_{CURR}. If no reply received, go to Step 17.

9: Search for an entry of MAC_{CURR} in SEC_{table}. If entry found, go to Step 13.

10: Search for an entry of MAC_{CURR} in $ALIAS_{table}$. If entry found, go to Step 12.

11: Store the mapping $MAC_{CURR} - IP_{CURR}$ in SEC_{table}. Go to Step 17.

12: Activate $ALARM_{MOD}$ by sending the message "MAC_{CURR} is causing IP Exhaustion problem". Go to Step 17.

13: Extract IP_{BCURR} from SEC_{table}. If (IP_{BCURR} = = IP_{CURR}), go to Step 17.

14: Send ARP_{REQ} to older mapping $MAC_{CURR} - IP_{BCURR}$.

15: If corresponding ARP_{REP} received, go to Step 12. If no reply received, go to Step 16.

16: Replace IP_{BCURR} with IP_{CURR} in SEC_{table}. Go to Step 17.

17: Move to next IP address in the range IP_{POSS}.

18: EXIT

The flowchart shown in Fig. 3.1 represents the new host entering algorithm.

3.3.1 Different Attack scenarios during execution of new host entering algorithm

1. *Flood of Spoofed ARP Replies:* We observed that if a flood of spoofed ARP Replies is sent, the legitimate response of probe ARP Request (ARP_{REQ}) gets dropped and does not reach to the host that has sent ARP_{REQ}. This is because the flood can be sent with maximum link bandwidth utilization. As a result, the host may consider that the older mapping is now invalid and the new incoming ARP_{REP} is a valid one. So the algorithm may update the SEC_{table} with a spoofed rogue entry. Moreover, we observed that the attack is effective only if attacker sends the flood; otherwise the algorithm will be able to receive the corresponding legitimate ARP_{REP}. To overcome this problem, we have added a sniffing functionality into the new host entering algorithm. The sniffer will sniff three packets within 500 milliseconds and it will check the type of packet. If all three packets are ARP messages, the sniffer will raise the alarm by activating $ALARM_{MOD}$ and sending the message "ARP flood detected by $MAC_{FLOODER}$ for $IP_{FLOODER}$" because in normal working conditions of the network, ARP messages are not generated so frequently. As a result, it is one of the effective signatures for detecting flood of ARP messages. So the attack is not possible while execution of new host entering algorithm.

2. *Attacker sniffs ARP_{REQ} and sends corresponding ARP_{REP}:* This is another attack scenario in which an attacker can sniff the ARP_{REQ} sent for IP_{CURR}

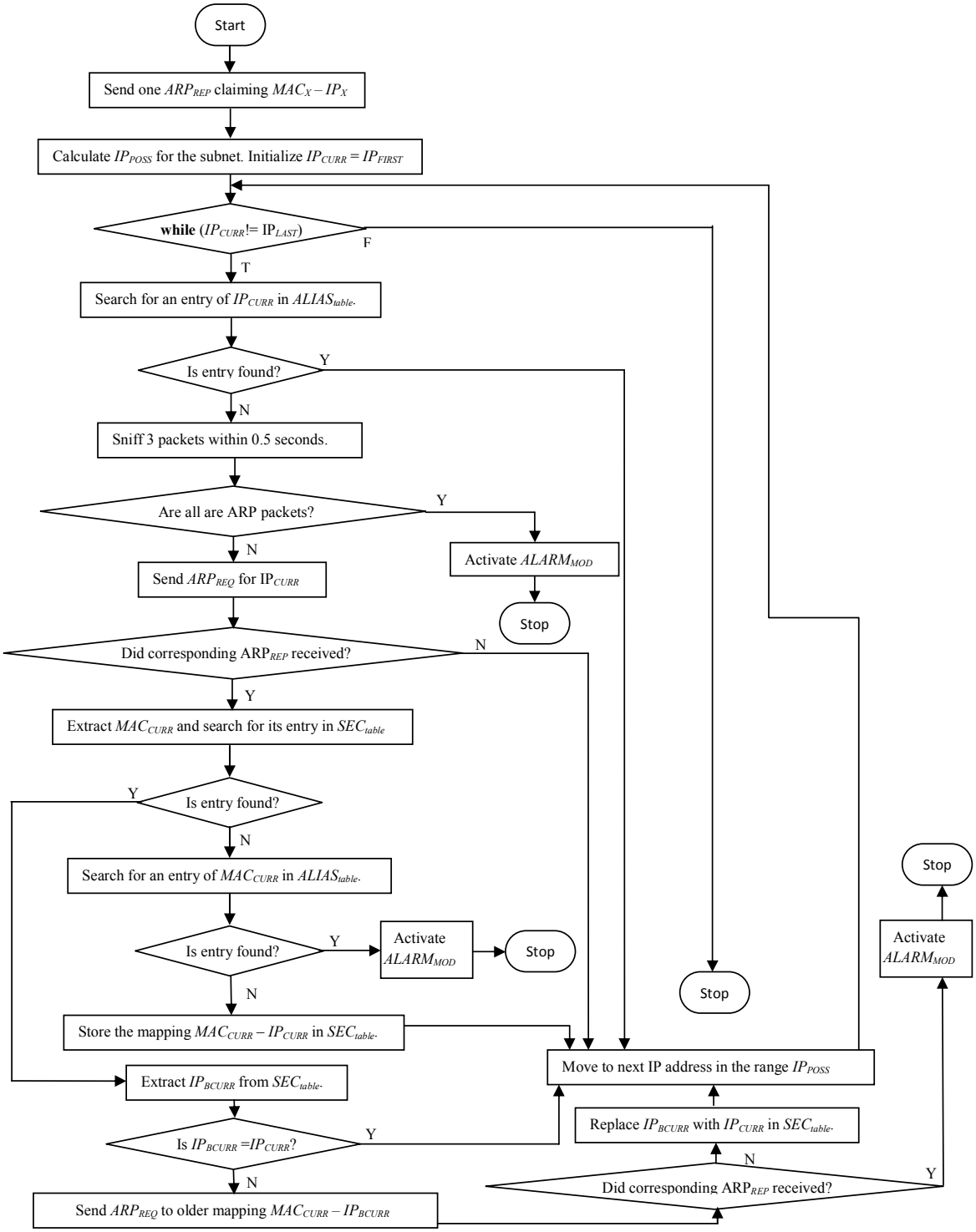


Fig. 3.1: Flowchart for the new host entering algorithm

and generate the corresponding spoofed ARP_{REP}. This spoofed ARP_{REP} sent by attacker may claim that IP_{CURR} is mapped to attacker's MAC address ($MAC_{attacker}$). However, we observed that it is the legitimate host (really possessing IP_{CURR}) whose ARP_{REP} is received first rather than that of attacker's. It is due to the fact that the attacker is first sniffing and then crafting the ARP_{REP} while ARP_{REP} came from the legitimate host is sent using lower level hardware support. Once ARP_{REP} from legitimate user is received, the algorithm will ignore the ARP_{REP} from attacker and move ahead for further execution. Thus, the attack will not be possible in this case also.

Suppose IP_{CURR} is not allotted to any of the hosts. Victim has sent an ARP_{REQ} to get the mapping of IP_{CURR} and then, attacker has sent the corresponding ARP_{REP}. As a result, the victim's new host entering algorithm will receive ARP_{REP} and search for $MAC_{attacker}$ into SEC_{table}. If an entry is found for $MAC_{attacker}$, the algorithm will validate that older mapping using probe packets. If reply does not come back, the older mapping will be deleted and the newer mapping will be entered into SEC_{table}. But if the reply comes back, the algorithm will raise the alarm by activating $ALARM_{MOD}$ and sending the message "$MAC_{attacker}$ is causing IP Exhaustion problem". Thus, the attack will be prevented. However if there is no entry for $MAC_{attacker}$ in SEC_{table}, the algorithm will search for $MAC_{attacker}$ in $ALIAS_{table}$. If such entry is found in $ALIAS_{table}$, the alarm will be raised. Otherwise, the victim will simply receive and store the mapping $MAC_{attacker} - IP_{CURR}$ into SEC_{table}. Suppose a new host enters into the network and configures itself with IP_{CURR} and sends a broadcasted ARP_{REP} claiming it possesses IP_{CURR}. Now, the victim's new host entering algorithm validate the older mapping ($MAC_{attacker} - IP_{CURR}$) using probe packets. If reply does not come back, the older mapping (attacker's mapping) will be deleted and the newer mapping (new host's mapping) will be made into SEC_{table}. However if reply comes back, it will detect that attack is going on and it will activate $ALARM_{MOD}$. Though the algorithm will not be able to know the real mapping for IP_{CURR}, it can still detect that attack has been launched.

3.4 EXISTING HOST ALGORITHM

Existing host algorithm will be continuously sniffing the ARP messages received by the host on which the algorithm is being executed. Its responsibility is to validate every ARP message, make entries into SEC_{table} and static entries into primary ARP cache, detect flood of spoofed ARP messages, activate $ALARM_{MOD}$ when ARP poisoning and IP Exhaustion attacks are detected. It executes along with the new host entering algorithm in parallel. Existing host algorithm uses ICMP probe packets instead of ARP probe packets to validate the ARP messages so as to make the algorithm more resistive against attacks. It will always be difficult for attacker to send spoofed ICMP replies rather than spoofed ARP Replies. We used ARP probe and ICMP probe in new host entering algorithm and existing host algorithm respectively so that if attacker wants to launch the attack, s/he has to generate spoofed replies for both the protocols. Also, attacker may be sometimes compelled to modify her/his network stack to handle probe ICMP echo requests.

When this algorithm will receive an ARP message, it will validate the message by searching for entries into the tables and then using probe packets. If the message is a genuine one, it will be store into SEC_{table} otherwise; the alarm module will be activated to raise the alarm corresponding to the type of attack. The flood mitigation functionality of existing host algorithm is same as that of new host entering algorithm. The example given below illustrates the working of existing host algorithm.

Working Example: In this example, we are assuming that there is no ARP flood sent by the attacker. We have discussed it while considering different attack scenarios in Section 3.4.1.

Suppose X is a host that is currently present into the network and executing existing host algorithm. We assume that its learning phase (execution of new host entering algorithm) is finished. The existing host algorithm is continuously sniffing the traffic so as to capture all ARP messages except the packets which is having source IP address or source MAC address of X, i.e. IP_X or MAC_X respectively. When the algorithm receives an ARP message (request/reply), it will extract the source IP address (IP_{SRC}) and the source MAC address (MAC_{SRC}). First, the

new host entering algorithm will search for the entry of IP_{SRC} into SEC_{table} and $ALIAS_{table}$ and then, it will search for the entry of MAC_{SRC} into both the tables. This can give rise to these situations:

1. *No Entry found for IP_{SRC} in SEC_{table}:* If there is no entry for IP_{SRC} in SEC_{table}, the algorithm will look for an entry of IP_{SRC} in $ALIAS_{table}$. This results into two cases:

   a. *No Entry found for IP_{SRC} in $ALIAS_{table}$:* If entry is not found in $ALIAS_{table}$ also, the final validation will be based on whether the entry for MAC_{SRC} is found in either of the tables or not.

   b. *Entry found for IP_{SRC} in $ALIAS_{table}$:* If entry for IP_{SRC} is found in $ALIAS_{table}$, the algorithm will extract the corresponding mapped MAC address (say, MAC_{BSRC}) from $ALIAS_{table}$. If MAC_{BSRC} and MAC_{SRC} are same, the algorithm will ignore the ARP_{REP}. But if both are different, the algorithm will activate the alarm module by sending the message "ARP Poisoning is detected for IP_{SRC}". Also, the existing host algorithm will make a static entry for the mapping $MAC_{BSRC} - IP_{SRC}$ into primary ARP cache of the victim so that the flood of spoofed ARP replies does not disrupt the network connectivity of the victim. Since the user has implicitly defined the mapping $MAC_{BSRC} - IP_{SRC}$ in $ALIAS_{table}$, it will be considered as valid by the algorithm and thus, the newer one will be considered as invalid.

2. *Entry found for IP_{SRC} in SEC_{table}:* If entry is found for IP_{SRC} in SEC_{table}, the algorithm will extract the corresponding mapped MAC address (say, MAC_{BSRC}) from SEC_{table}. If MAC_{BSRC} and MAC_{SRC} are same, the algorithm will ignore the ARP_{REP}. But if both are different, the algorithm will craft and send five probe ICMP echo requests for the older mapping $MAC_{BSRC} - IP_{SRC}$. If the response comes back for at least one of the probing packets, it means that the older mapping is still valid while the newer one is a spoofed ARP message because one IP address can be allotted to a single MAC address at a time. Thus the $ALARM_{MOD}$ will be

activated by sending the message "ARP Poisoning is detected for IP_{SRC}". Also, the existing host algorithm will make a static entry for the mapping $MAC_{BSRC} - IP_{SRC}$ into primary ARP cache of the victim. However, if response does not come back for any of the probe packets, the algorithm will consider that the older mapping does not exist now. So, the algorithm will send a probe ICMP echo request for the newer mapping $MAC_{SRC} - IP_{SRC}$. If the corresponding probe reply is received, the algorithm will replace older mapping $MAC_{BSRC} - IP_{SRC}$ with newer $MAC_{SRC} - IP_{SRC}$ mapping. But if the corresponding probe reply is not received for newer mapping also, the $ALARM_{MOD}$ will be activated by sending the message "ARP Poisoning is detected for IP_{SRC}".

The algorithm has sent five probe packets so that in case of flood (at slower speeds), the probe sending host can receive at least one response. Even if flood is launched with maximum link utilization also, the flood detecting functionality of the existing host algorithm will immediately detect the launch of flood.

3. *No Entry found for MAC_{SRC} in SEC_{table}:* If there is no entry for MAC_{SRC} in SEC_{table}, the algorithm will look for an entry of MAC_{SRC} in $ALIAS_{table}$. This results into two cases:

   a. *No Entry found for MAC_{SRC} in $ALIAS_{table}$:* If entry is not found in $ALIAS_{table}$ also, the algorithm will send a probe ICMP echo request for the received ARP message, i.e. mapping of $MAC_{SRC} - IP_{SRC}$. If the corresponding probe reply is received, the algorithm will store the $MAC_{SRC} - IP_{SRC}$ mapping into SEC_{table}.

   b. *Entry found for MAC_{SRC} in $ALIAS_{table}$:* If entry is found in $ALIAS_{table}$, the algorithm will extract the mapped IP address (say, IP_{BSRC}) from $ALIAS_{table}$. If IP_{BSRC} and IP_{SRC} are same, the algorithm will ignore the ARP_{REP}. But if both are different, the algorithm will activate $ALARM_{MOD}$ by sending the message "Spoofed ARP Reply from MAC_{SRC}". Also, the existing host algorithm will make a static

entry for the mapping $MAC_{SRC} - IP_{BSRC}$ into primary ARP cache of the victim.

4. *Entry found for MAC_{SRC} in SEC_{table}:* If entry is found in SEC_{table}, the algorithm will extract the corresponding mapped IP address (say, IP_{BSRC}). After this, the algorithm will send five probe ICMP echo requests for the mapping $MAC_{SRC} - IP_{BSRC}$. If at least one corresponding probe reply is received, it means that the older mapping is still valid while the newer one is a spoofed ARP message because one MAC address can have only one IP address at a time (until and unless IP aliasing is configured for that MAC address). Thus the $ALARM_{MOD}$ will be activated by sending the message "MAC_{SRC} is creating IP Exhaustion Problem". However, if response does not come back for any of the probe packets, the algorithm will consider that the older mapping does not exist now. So, the algorithm will send a probe ICMP echo request for the newer mapping $MAC_{SRC} - IP_{SRC}$. If the corresponding probe reply is received, the algorithm will replace older mapping $MAC_{SRC} - IP_{BSRC}$ with newer $MAC_{SRC} - IP_{SRC}$ mapping. But if the corresponding probe reply is not received for newer mapping also, the $ALARM_{MOD}$ will be activated by sending the message "MAC address MAC_{SRC} is causing IP Exhaustion problem".

Few more short notations used in the existing host algorithm are as follows:

IP_{SRC} - Source IP address of host generating ARP message; MAC_{SRC} - Source MAC address of host generating ARP message; $ICMP_{REQ}$ - Probe ICMP Echo Request; $ICMP_{REP}$ - Probe ICMP Echo Reply;

Formal Discussion of existing host Algorithm executing at host X is discussed below:

Algorithm 2: **Existing host Algorithm**

Input: ARP Message (Request/Reply)

Output: Validating ARP messages. If validation passed, store MAC - IP bindings into SEC_{table} (if not present). If validation failed, activate $ALARM_{MOD}$.

1: Sniff the traffic and capture ARP messages.

2: Sniff 3 packets from the network within 0.5 seconds. If all 3 packets are not ARP packets, go to Step 4.

3: Extract $MAC_{FLOODER}$ and $IP_{FLOODER}$. Activate $ALARM_{MOD}$ by sending the message "ARP flood detected by $MAC_{FLOODER}$ for $IP_{FLOODER}$"

4: If ((IP_{SRC} != IP_X) or (MAC_{SRC} != MAC_X)), go to Step 5. Else, go to Step 1.

5: Search for an entry of IP_{SRC} in SEC_{table}. If no entry found, go to Step 13.

6: Extract MAC_{BSRC} from SEC_{table}. If (MAC_{BSRC} ! = MAC_{SRC}), go to Step 8.

7: Ignore ARP message and go to Step 1.

8: Send five $ICMP_{REQ}$ for $MAC_{BSRC} - IP_{SRC}$. If not a single $ICMP_{REP}$ received, go to Step 10.

9: Activate $ALARM_{MOD}$ by sending the message "ARP Poisoning is detected for IP_{SRC}". Make static entry of $MAC_{BSRC} - IP_{SRC}$ into primary ARP cache. Go to Step 1.

10: Send an $ICMP_{REQ}$ for $MAC_{SRC} - IP_{SRC}$. If no corresponding $ICMP_{REP}$ received, go to Step 12.

11: Replace $MAC_{BSRC} - IP_{SRC}$ mapping with $MAC_{SRC} - IP_{SRC}$ mapping. Go to Step 1.

12: Activate $ALARM_{MOD}$ by sending the message "ARP Poisoning is detected for IP_{SRC}". Go to Step 1.

13: Search for an entry of IP_{SRC} in $ALIAS_{table}$. If no entry found, go to Step 17.

14: Extract MAC_{BSRC} from $ALIAS_{table}$. If (MAC_{BSRC} ! = MAC_{SRC}), go to Step 16.

15: Ignore ARP message and go to Step 1.

16: Activate $ALARM_{MOD}$ by sending the message "ARP Poisoning is detected for IP_{SRC}". Make static entry of $MAC_{BSRC} - IP_{SRC}$ into primary ARP cache. Go to Step 1.

17: Search for an entry of MAC_{SRC} in SEC_{table}. If no entry found, go to Step 24.

18: Extract IP_{BSRC} from SEC_{table}.

19: Send five $ICMP_{REQ}$ for $MAC_{SRC} - IP_{BSRC}$. If not a single $ICMP_{REP}$ received, go to Step 21.

20: Activate $ALARM_{MOD}$ by sending the message "MAC_{SRC} is creating IP Exhaustion problem". Make static entry of $MAC_{BSRC} - IP_{SRC}$ into primary ARP cache. Go to Step 1.

21: Send an $ICMP_{REQ}$ for $MAC_{SRC} - IP_{SRC}$. If no corresponding $ICMP_{REP}$ received, go to 23.

22: Replace $MAC_{BSRC} - IP_{SRC}$ mapping with $MAC_{SRC} - IP_{SRC}$ mapping. Go to Step 1.

23: Activate $ALARM_{MOD}$ by sending the message "MAC_{SRC} is creating IP Exhaustion problem". Go to Step 1.

24: Search for an entry of MAC_{SRC} in $ALIAS_{table}$. If no entry found, go to Step 28.

25: Extract IP_{BSRC} from $ALIAS_{table}$. If ($IP_{BSRC} != IP_{SRC}$), go to Step 27.

26: Ignore ARP message and go to Step 1.

27: Activate $ALARM_{MOD}$ by sending the message "Spoofed ARP Reply from MAC_{SRC}". Make static entry of $MAC_{SRC} - IP_{BSRC}$ into primary ARP cache. Go to Step 1.

28: Send an $ICMP_{REQ}$ for $MAC_{SRC} - IP_{SRC}$. If corresponding $ICMP_{REP}$ received, store this binding into SEC_{table}. Go to Step 1.

29: Ignore ARP message and go to Step 1.

30: EXIT

The flowchart shown in Fig. 3.2 represents the existing host algorithm.

3.4.1 Different Attack scenarios during execution of existing host algorithm

1. *Flood of Spoofed ARP Replies:* Like new host entering algorithm, the existing host algorithm also detects the flood of spoofed ARP Replies in the same way. So, we need not explain this scenario for existing host algorithm.

2. *Attacker sends spoofed ARP message when corresponding mapping is not present in both the tables:* Suppose attacker sends an ARP message with mapping $MAC_Y - IP_Y$. When a host (say, X) will receive this ARP message, the existing host algorithm will search for the MAC_Y and IP_Y entries into SEC_{table} and $ALIAS_{table}$. Since, X's tables do not have this entry; the existing host algorithm will send $ICMP_{REQ}$ for this mapping. Now if attacker can send corresponding spoofed $ICMP_{REP}$ also either by modifying her/his protocol stack or through any other means, X's existing host algorithm will simply store the mapping $MAC_Y - IP_Y$ into SEC_{table}. But when a host will send an ARP message which has really been allotted IP_Y, X's existing host algorithm will then send five $ICMP_{REQ}$ for the previous mapping $MAC_Y - IP_Y$ (sent by attacker). If attacker again sends the corresponding $ICMP_{REP}$, X's existing host algorithm will activate $ALARM_{MOD}$. Moreover, if attacker does not send the corresponding $ICMP_{REP}$, X's existing host algorithm will simply replace the previous mapping sent by the attacker with the newer mapping sent by the genuine host. So the attack will not be possible.

3. *Attacker sends spoofed ARP message when corresponding mapping is present in either of the tables:* Suppose attacker sends an ARP message with mapping $MAC_Y - IP_Y$ such that one of the tables (SEC_{table} or $ALIAS_{table}$) at host X is already having an entry for MAC_Y or IP_Y. When X will receive this mapping, it will search its table for the entries. Since the entry is already

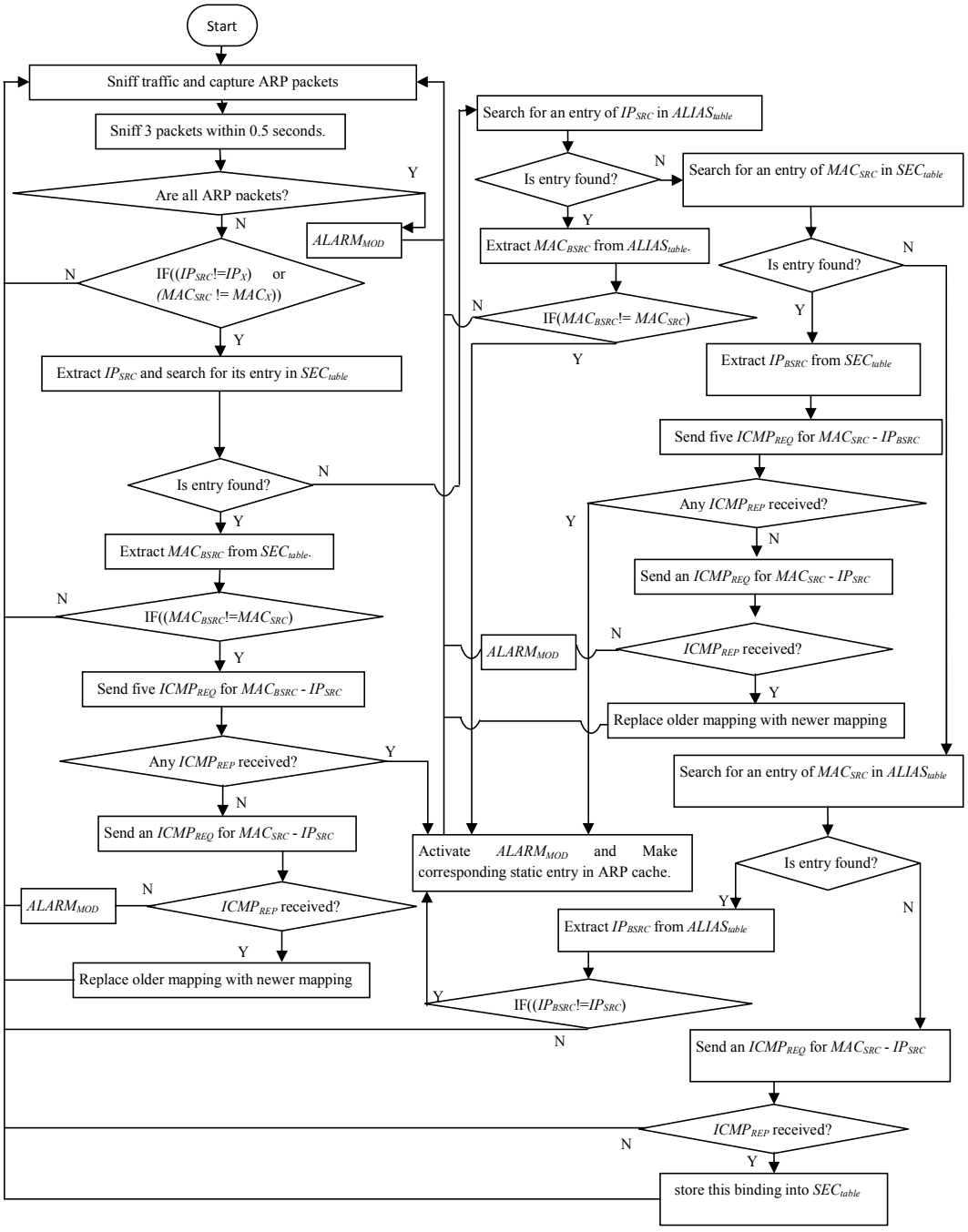


Fig. 3.2: Flowchart for the Existing host algorithm

present, the existing host algorithm will send $ICMP_{REQ}$ for the older mapping. When the algorithm will receive the corresponding $ICMP_{REP}$ back, it will activate $ALARM_{MOD}$. Also, the algorithm will create a static entry for older mapping into primary cache. So the attack will not be possible in this scenario also.

4. *Attacker trying to get MITM position:* A situation may arise where an attacker tries to get MITM position between two communicating entities. It may involve communication either between one host and the Default Gateway (DG) or between two hosts in which one may be a server. We divided this scenario further depending on the type of entities involved in the communication. The different scenarios are:

    a. *Attack against DG and host X:* If an attacker wants to get MITM position between host X and DG, s/he has to poison the ARP cache of both the entities. If X is a new host that has just entered into the network, attacker can easily claim itself as DG by sending fake ARP_{REP} to X because X is not having an entry for DG's MAC address or DG's IP address. Similarly, s/he can poison DG's cache also because it is not having any entry for IP_X or MAC_X. However when DG will receive ARP message from X or vice versa, they will get to know each other's mapping. So immediately the $ALARM_{MOD}$ will be activated. It is also possible that DG is having an entry for attacker's IP address or MAC address. In such case, DG will activate $ALARM_{MOD}$ as soon as it receives spoofed ARP message (having different mapping) from attacker. So, attack will be prevented.

    b. *Attack against host X and host Z:* In this scenario also, the attacker can perform the attack in similar way but as soon as the hosts will learn each other's mapping, the attack will be detected and prevented. Here also, it is possible that both the hosts are having an entry for attacker's IP address or MAC address. In such case, they will activate $ALARM_{MOD}$ as soon as they receive spoofed ARP message (having different mapping)

from attacker.

3.5 ALARM MODULE

Alarm Module is integrated with the proposed scheme so that the network administrators can get the alert when attacker tries to launch the attack. Depending upon the type of attack, the algorithms send different messages to activate the alarm module. Our alarm module first writes the message sent by the algorithms on a text file and then this text is converted into speech using a speech synthesis tool. For our purpose, we used Festival Speech Synthesis Tool [23]. This tool will synthesize the text written by the alarm module into corresponding speech. Also, we used an alarm siren which is played using a utility mpg123 [24]. The utility plays the files on command line interface. Overall, this alarm module provides an effective alerting mechanism for network administrator.

CHAPTER 4
EXPERIMENTAL RESULTS AND ANALYSIS

4.1 EXPERIMENTAL SETUP

We have implemented the scheme within a test LAN having more than 100 hosts live at a time. To demonstrate the attack scenarios and the working of the proposed scheme, we have allotted three of these hosts as Default Gateway (DG), host X (victim) and host Y (attacker). Both X and Y is using Kali Linux operating system [25]. Table 4.1 shows the IP and MAC addresses of these hosts.

4.2 EXPERIMENTAL RESULTS

In this section, we have shown the practical implementation of IP Exhaustion and classical ARP poisoning attack which is generally used to get MITM position between two communicating entities. We have also shown the detection and prevention of these attacks using the proposed scheme.

Fig. 4.1 shows the normal execution of entering algorithm at X. The algorithm started sending probe ARP requests to other hosts and when corresponding replies received, the mappings are stored into SEC_{table}. The normal execution of existing host algorithm at X is shown in Fig. 4.2. The existing host algorithm is continuously

Table 4.1: IP Addresses and MAC Addresses of different hosts

Host	IP Address	MAC Address
Gateway(DG)	10.7.6.1 (IP_{DG})	00:00:cd:2b:d2:f7 (MAC_{DG})
Victim(X)	10.7.6.134 (IP_X)	00:26:22:04:50:2a (MAC_X)
Attacker(Y)	10.7.6.113 (IP_Y)	1c:75:08:49:d8:d1 (MAC_Y)

```
root@NIKHIL:~/Desktop/1402idrbt# python send_ping_for_secache.py 10.7.6.0/24
WARNING: No route found for IPv6 destination :: (no default route?)
Broadcasting your MAC-IP mapping into the network
.
Sent 1 packets.
------------------------------------------------------------------
Sending ARP Request for IP 10.7.6.0
fail 1: Ether / ARP who has 10.7.6.0 says 10.7.6.186

Sent 1 packets, received 0 packets. 0.0% hits.
Sending ARP Request for IP 10.7.6.1
RECV 1: Ether / ARP is at 00:00:cd:2b:d2:f7 says 10.7.6.1 / Padding

Sent 1 packets, received 1 packets. 100.0% hits.
ARP Reply received
Storing the mapping into Secondary table
Sending ARP Request for IP 10.7.6.2
fail 1: Ether / ARP who has 10.7.6.2 says 10.7.6.186

Sent 1 packets, received 0 packets. 0.0% hits.
```

Fig. 4.1: New host entering algorithm execution without launching attack

receiving and validating ARP messages. As a result, the genuine mappings are stored in SEC_{table} if it is not present initially. The entries in primary ARP cache of X (before Y launches the attack) are shown in Fig. 4.3. In normal scenario, after ARP poisoning, the ARP cache of X is changed and hence, shown in Fig. 4.4.

It may be possible that an attacker can launch IP Exhaustion attack when the new host entering algorithm at X is in learning phase. The new host entering algorithm at X sends a probe ARP request for an IP address and attacker has sent a spoofed corresponding ARP reply for that IP address. When the new host entering algorithm received this reply, it ensured that the MAC address of received ARP Reply should not be present in SEC_{table} or $ALIAS_{table}$. Since entry for the MAC_Y was already present in SEC_{table}, the new host entering algorithm sent a probe ARP request for older mapping. Since the algorithm received the corresponding reply back, it activated $ALARM_{MOD}$ by sending message "MAC_Y is creating IP Exhaustion problem". It is shown in Fig. 4.5. An attacker may also send a flood of spoofed ARP messages. The new host entering algorithm can detect it effectively. It is shown in Fig. 4.6.

Fig. 4.7 shows the execution of existing host algorithm at X when Y has sent a spoofed ARP message claiming IP_{DG} belongs to MAC_Y. Since SEC_{table} already contained the entry for IP_{DG}, the existing host algorithm sent an ICMP probe request for the older mapping. When it came to know that older mapping still exists,

```
root@NIKHIL:~/Desktop/0503idrbt# python existing.py 10.7.6.134 00:26:22:04:50:2a
WARNING: No route found for IPv6 destination :: (no default route?)
ARP message received
Entry not present in secondary table.Sending ICMP probe for ARP message
fail 1: Ether / IP / ICMP 10.7.6.186 > 10.7.6.97 echo-request 0

Sent 1 packets, received 0 packets. 0.0% hits.
ICMP Reply not received.Ignore the ARP message
ARP message received
Entry not present in secondary table.Sending ICMP probe for ARP message
RECV 1: Ether / IP / ICMP 10.7.6.155 > 10.7.6.186 echo-reply 0 / Padding

Sent 1 packets, received 1 packets. 100.0% hits.
ICMP Reply received. Storing the mapping into secondary table
ARP message received
IP entry found in secondary table
Same MAC-IP mapping already present
Entry not present in secondary table.Sending ICMP probe for ARP message
fail 1: Ether / IP / ICMP 10.7.6.186 > 10.7.6.186 echo-request 0

Sent 1 packets, received 0 packets. 0.0% hits.
ICMP Reply not received.Ignore the ARP message
```

Fig. 4.2: Existing host algorithm execution without launching attack

```
root@NIKHIL:~/Desktop/0503idrbt# arp -a
? (10.7.6.1) at 00:00:cd:2b:d2:f7 [ether] on eth0
? (10.7.6.113) at 1c:75:08:49:d8:d1 [ether] on eth0
root@NIKHIL:~/Desktop/0503idrbt# _
```

Fig. 4.3: X's primary ARP cache before attack

```
root@NIKHIL:~/Desktop/0503idrbt# arp -a
? (10.7.6.1) at 1c:75:08:49:d8:d1 [ether] on eth0
? (10.7.6.122) at 2c:27:d7:bc:21:18 [ether] on eth0
? (10.7.6.113) at 1c:75:08:49:d8:d1 [ether] on eth0
root@NIKHIL:~/Desktop/0503idrbt# _
```

Fig. 4.4: X's primary ARP cache after attack

```
ARP Reply received
MAC entry found in secondary table
Sending probe ARP Request for older mapping
RECV 1: Ether / ARP is at 1c:75:08:49:d8:d1 says 10.7.6.113 / Padding

Sent 1 packets, received 1 packets. 100.0% hits.
Received probe ARP reply for older mapping
MAC address 1c:75:08:49:d8:d1 is causing IP Exhaustion Problem
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.14.4; written and copyright by Michael Hipp and others
        free software (LGPL/GPL) without any warranty but with best wishes

Playing MPEG stream 1 of 1: alarm.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 48000 Hz joint-stereo

[0:04] Decoding of alarm.mp3 finished.
```

Fig. 4.5: New host entering algorithm's execution while IP Exhaustion attack is going on

```
Sent 1 packets, received 0 packets. 0.0% hits.
Sending ARP Request for IP 10.7.6.3
fail 1: Ether / ARP who has 10.7.6.3 says 10.7.6.134

Sent 1 packets, received 0 packets. 0.0% hits.
Flood Detected from IP 10.7.6.1 and MAC 1c:75:08:49:d8:d1
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.14.4; written and copyright by Michael Hipp and others
        free software (LGPL/GPL) without any warranty but with best wishes

Playing MPEG stream 1 of 1: alarm.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 48000 Hz joint-stereo

[0:04] Decoding of alarm.mp3 finished.
```

Fig. 4.6: New host entering algorithm's execution while attacker launches the flood

```
ARP message received
IP entry found in secondary table
Mapped MAC is not the same
Sending ICMP probe request for older mapping
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.1 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.1 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.1 echo-request 0
RECV 1: Ether / IP / ICMP 10.7.6.1 > 10.7.6.134 echo-reply 0 / Padding
RECV 1: Ether / IP / ICMP 10.7.6.1 > 10.7.6.134 echo-reply 0 / Padding

Sent 5 packets, received 2 packets. 40.0% hits.
ICMP reply received for older mapping
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.14.4; written and copyright by Michael Hipp and others
        free software (LGPL/GPL) without any warranty but with best wishes

Playing MPEG stream 1 of 1: alarm.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 48000 Hz joint-stereo

[0:04] Decoding of alarm.mp3 finished.
```

Fig. 4.7: Existing host algorithm's execution while ARP poisoning attack is going on

```
root@NIKHIL:~/Desktop/0503idrbt# arp -a
? (10.7.6.1) at 00:00:cd:2b:d2:f7 [ether] PERM on eth0
? (10.7.6.122) at 2c:27:d7:bc:21:18 [ether] on eth0
root@NIKHIL:~/Desktop/0503idrbt# _
```

Fig. 4.8: Static entry is made into X's ARP cache to prevent the attack further

it activated $ALARM_{MOD}$ by sending message "ARP poisoning is detected for IP address IP_{DG}". Also, the algorithm made static entry for IP_{DG} into primary ARP cache of X. It is shown in Fig. 4.8. An attacker may launch an IP Exhaustion attack as well. Fig. 4.9a shows IP Exhaustion attack launched by Y without modifying her/his network stack. Y has sent an ARP message to X claiming that "10.7.6.60" belongs to MAC_Y. When X's existing host algorithm sent ICMP probe requests to older mapping ($IP_Y - MAC_Y$), it found out that the older mapping is still valid. So, $ALARM_{MOD}$ got activated. Also, Y can modify its stack in such a way that it does not reply for older IP (IP_Y) while it replies for newer IP (10.7.6.60). In this case, X's existing host algorithm will consider the newer mapping as genuine one so the existing host algorithm will replace the older mapping with newer one into SEC_{table} to detect and prevent the attack. It is shown in Fig. 4.9b. However, when a host with real IP (10.7.6.60) will enter into the network, the existing host algorithm will detect the attack immediately.

```
ARP message received
MAC entry found in secondary table.
Mapped IP is not the same
Sending ICMP probe request for older mapping
RECV 1: Ether / IP / ICMP 10.7.6.113 > 10.7.6.134 echo-reply 0 / Padding
RECV 1: Ether / IP / ICMP 10.7.6.113 > 10.7.6.134 echo-reply 0 / Padding
RECV 1: Ether / IP / ICMP 10.7.6.113 > 10.7.6.134 echo-reply 0 / Padding
RECV 1: Ether / IP / ICMP 10.7.6.113 > 10.7.6.134 echo-reply 0 / Padding
RECV 1: Ether / IP / ICMP 10.7.6.113 > 10.7.6.134 echo-reply 0 / Padding

Sent 5 packets, received 5 packets. 100.0% hits.
ICMP reply received for older mapping
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.14.4; written and copyright by Michael Hipp and others
        free software (LGPL/GPL) without any warranty but with best wishes

Playing MPEG stream 1 of 1: alarm.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 48000 Hz joint-stereo

[0:04] Decoding of alarm.mp3 finished.
```

(a)

```
ARP message received
MAC entry found in secondary table.Mapped IP is not the same
Sending ICMP probe request for older mapping
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.113 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.113 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.113 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.113 echo-request 0
fail 1: Ether / IP / ICMP 10.7.6.134 > 10.7.6.113 echo-request 0

Sent 5 packets, received 0 packets. 0.0% hits.
Sending ICMP probe request for newer mapping
RECV 1: Ether / IP / ICMP 10.7.6.60 > 10.7.6.134 echo-reply 0 / Padding

Sent 1 packets, received 1 packets. 100.0% hits.
ICMP reply not received for older mapping
ICMP reply received for newer mapping.Replace older one
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.14.4; written and copyright by Michael Hipp and others
        free software (LGPL/GPL) without any warranty but with best wishes

Playing MPEG stream 1 of 1: alarm.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 48000 Hz j▯int-stereo

[0:04] Decoding of alarm.mp3 finished.
```

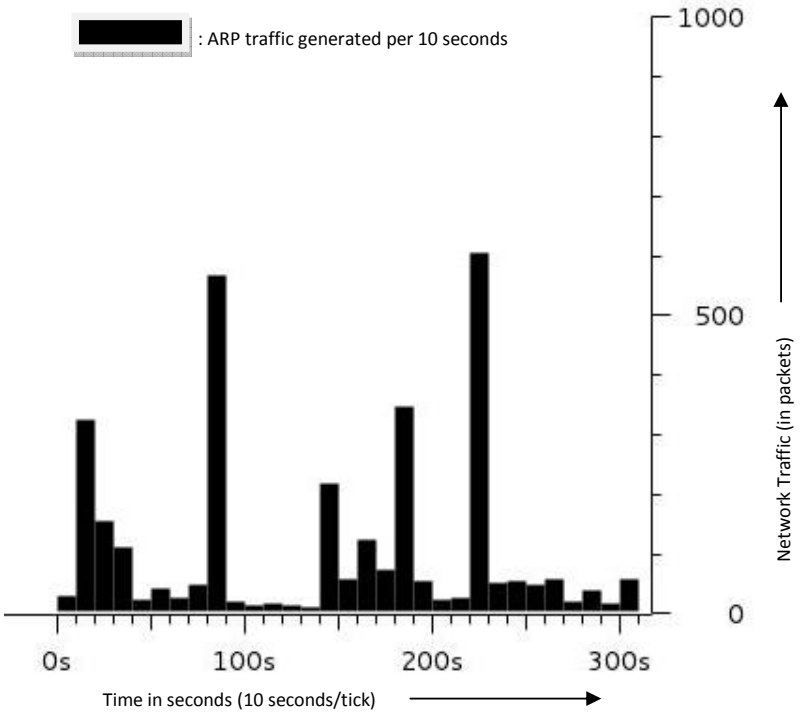
(b)

Fig. 4.9: (a) Existing host algorithm's execution while IP Exhaustion attack is going on with attacker's network stack unchanged, (b) Existing host algorithm's execution while IP Exhaustion attack is going on with attacker's network stack changed

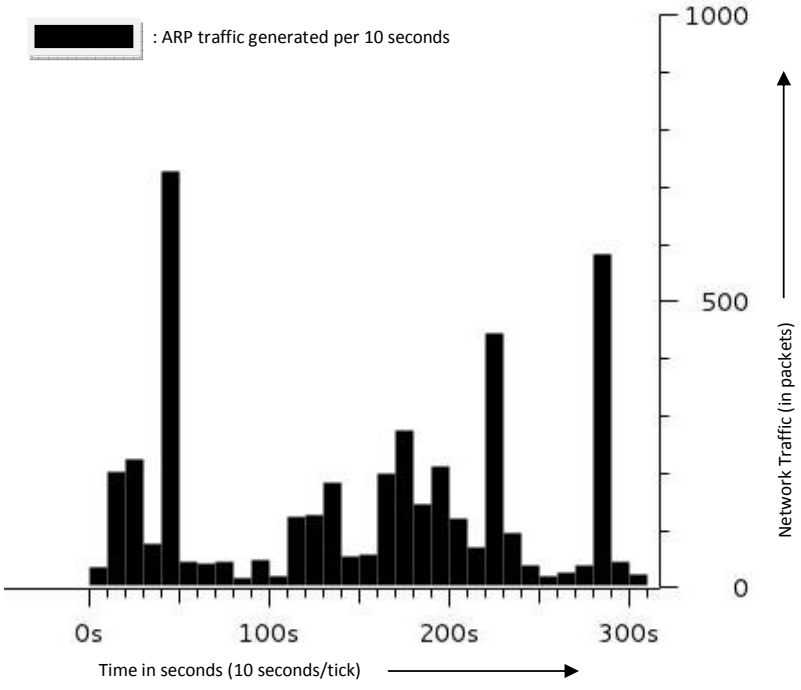
4.3 TRAFFIC ANALYSIS

We have used network analyzer Wireshark [26] to capture the network behavior during the absence and presence of proposed scheme implementation. Fig. 4.10a shows the ARP traffic within the network when the new host entering algorithm is not executing. The Y - axis represents the number of packets generated. The X - axis represents the time with a single tick of 10 seconds. The black bars represent the ARP traffic generated per 10 seconds. The traffic is analyzed for about 300 seconds. Similarly, Fig. 4.10b shows the network behavior during the execution of new host entering algorithm. We can see that the black bars in both the figures are almost same. So, we can say that the ARP traffic generated by the new host entering algorithm is not causing much overhead on the network.

Fig. 4.11a shows the ARP and ICMP traffic during the normal working of network without the execution of existing host algorithm. The Y - axis represents the number of packets generated. The X - axis represents the time with a single tick of 10 seconds. The peaks represents the generated ARP traffic while the red thick dots represent the ICMP traffic. Similarly, Fig. 4.11b shows the network behavior during the execution of existing host algorithm. We can see that the peaks in both the figures are almost same. However, the number of red thick dots are increased in Fig. 4.11b as compared to Fig. 4.11a. Thus, we can say that the ARP traffic generated by the existing host algorithm is not causing much overhead on the network but the ICMP traffic is increased due to the execution of the existing host algorithm.

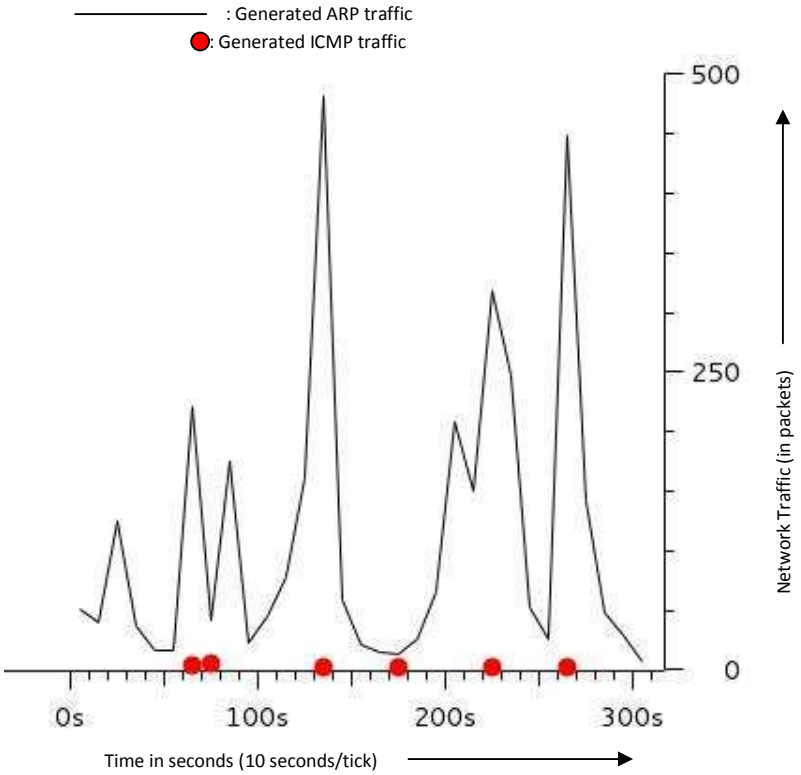


(a)

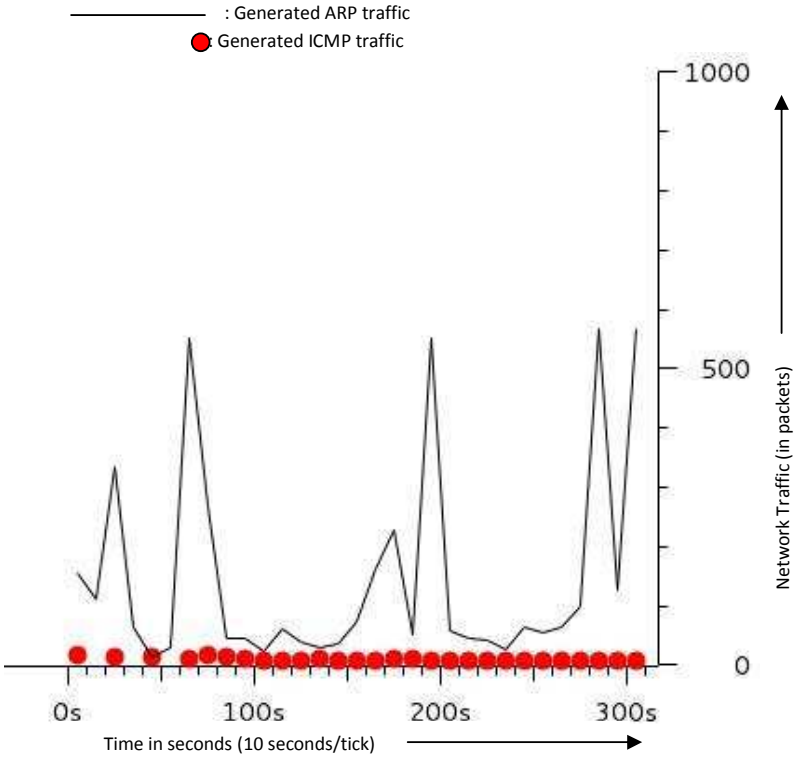


(b)

Fig. 4.10: (a) ARP traffic within network during normal conditions, (b) ARP Traffic while new host entering algorithm's execution is going on



(a)



(b)

Fig. 4.11: (a) ARP and ICMP Traffic within network during normal conditions, (b) ARP and ICMP Traffic while existing host algorithm's execution is going on

CHAPTER 5

COMPARISON OF OUR PROPOSED SCHEME WITH OTHER SCHEMES

We have considered five different parameters to compare the previously proposed schemes. They are:

1. Flood of spoofed ARP messages.
2. IP Exhaustion problem.
3. Backward compatibility with the existing network infrastructure.
4. Single point of failure problem.
5. Compatibility with the IP aliasing configurations.

We have discussed each of these parameters in the following section.

5.1 FLOOD OF SPOOFED ARP MESSAGES

Various proposed solutions in the literature use probing process to mitigate ARP poisoning attacks. The probing process involves the usage of ARP/ICMP messages to validate the ARP messages received by them. These probe packets (ARP/ICMP messages) are sent when an ARP message is received by a host.

Suppose A is a host which uses probing process. Now, C sends an ARP message to A saying B's IP address, IP (B) belongs to C's MAC address, MAC (C). When A will receive this ARP message, it will transmit back a broadcasted ARP probe request, stating, "Who is having IP (B)?". This ARP request will be received by both, B and C. Since B really possesses IP (B), it will respond to the ARP Request. On the other hand, if C wants to conceal his/her individuality, s/he has to reply to

all queries for IP (B). Now A will get to know that C may have sent a spoofed ARP message because two different MAC addresses can not possess same IP address at a time. Therefore, A can raise an alert saying "ARP Poisoning is detected for IP (B)".

However, if C sends a flood of spoofed ARP replies saying "IP (B) belongs to MAC (C)" with a speed of maximum link-utilization, A will not be able to get B's response. This is due to the fact that the C's spoofed flood will overwhelm the B's real response. As a result, B's response will be dropped. Ultimately, in this case, A will think that B has gone offline and IP (B) has been allotted to MAC (C). This spoofed mapping, IP (B) - MAC (C), will be stored in A's primary cache which finally leads to ARP poisoning attack.

There are many tools and customized packet generators which are used to craft packets with custom headers. These tools even allow the attackers to specify the speed with which the packets are to be sent. One among these tools are packEth [27].

5.2 IP EXHAUSTION PROBLEM

Some of the proposed solutions within the literature consider that if a MAC address, MAC (X), is the first one to claim that an IP address, IP (P), belongs to it, IP (P) - MAC (X) mapping is a genuine one. Thereafter, the subsequent ARP messages claiming IP (P) belong to any other MAC address will be considered as the fake one.

The IP Exhaustion problem is essentially a type of ARP poisoning attack. The attacker broadcasts multiple ARP messages on the behalf of all unused (i.e. Not alive) IP addresses in the subnet. The attacker claims that all the unused IP addresses inside the subnet belong to my MAC address. When the other hosts receive this message, they update their primary ARP cache accordingly.

Suppose that hosts inside the subnet are executing the above mentioned scheme. If an attacker launches the IP Exhaustion attack, the hosts will update their cache accordingly. Now, hosts will assume that all the possible IP addresses inside the subnet are currently in use. So whenever, a genuine host will come into the network and it will send an ARP message, all other hosts will consider the new host as

illegitimate one. Finally, a genuine host may get banned while on the other hand, the attacker can enjoy total exemption within the network.

5.3 BACKWARD COMPATIBILITY WITH THE EXISTING NETWORK INFRASTRUCTURE

Several proposed schemes require a modification in the existing ARP specification. The required changes include the integration of cryptographic schemes with the ARP specifications.

However, the modification of ARP specifications will make the protocol incompatible with the existing network infrastructure. Though these types of strategies are rather efficient, they are not that much popular in the real world implementation because of the cost to be incurred.

5.4 SINGLE POINT OF FAILURE PROBLEM

Some of the proposed solutions in the literature suffer from the problem of single point of failure. This is due to their centralized nature. Most of such solutions are based on a detection server. The detection server is responsible for validating the ARP entries present in the primary ARP cache of all the hosts within the subnet.

However, if the server goes down, the network becomes insecure. In such instances, the attackers can easily establish the attack without the concern of becoming arrested.

5.5 COMPATIBILITY WITH THE IP ALIASING CONFIGURATIONS

Sometimes, the network administrators configure IP aliasing for some of the MAC addresses. Using IP Aliasing configurations, the network administrator can allot more than one IP addresses to a single MAC address. However, when such configurations are present within the LAN, the proposed schemes may give false alarms.

Suppose that IP aliasing is configured for MAC (A) and it possesses IP (X) and IP (Y) at the same time. Now, when two ARP messages with different source IP addresses (i.e. IP (X) and IP (Y)) but same MAC address (MAC (A)) will be

received by other hosts, they will raise the alarm saying IP Exhaustion attack is launched by MAC (A). But, in reality, the host with an interface (MAC (A)) is a genuine one. So, this limitation results in execution of false alarms.

5.6 VARIOUS PROPOSED SCHEMES IN THE LITERATURE

In this section, we have discussed some of the proposed schemes. Along with that, we have also presented their behavior against the comparative parameters which are talked about in the previous section.Following eight schemes are compared with respect to the five parameters which are discussed in the previous section:

1. Cryptography based schemes.
2. Passive Detection.
3. Kernel based Patches.
4. Compatibility with IP Aliasing configurations.
5. Usage of ICMP packets as probe packets.
6. Using Host based Discrete Event System
7. ICMP based secondary cache approach
8. Our Proposed Scheme (ARP - ICMP probe packet based approach)

5.6.1 Cryptography based schemes

Different modifications in ARP specifications are proposed in some of the schemes like Secure ARP [12] and Ticket based ARP [13].

Secure ARP is based on the concept of public/private key certificates. These digital certificates are used for the authentication of every ARP replies within the network. The Authoritative Key Distributor (AKD) acts as a central server to distribute the public keys to different hosts. This scheme makes use of Secure-DHCP (Dynamic Host Configuration Protocol) instead of usual DHCP.

Ticket based ARP involves the use of a ticket with each ARP message. Local Ticket Agent (LTA) is responsible for the distribution of this ticket.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since this scheme is not based on the probing process, it is not vulnerable to flooding of spoofed ARP messages.
- **Resistant to IP Exhaustion problem:** Since the digital certificates and tickets are used to validate every ARP reply within the LAN, the attacker cannot send spoofed ARP messages. Therefore, the scheme is effective against IP Exhaustion attack also.
- **Backward compatibility with the existing network infrastructure:** Due to such high level implementations, this scheme requires changes in the ARP standard specification. As a consequence, this scheme is not compatible with the existing networks.
- **Single point of failure problem:** Since the scheme involves the use of centralized entities like AKD server and LTA agent, the scheme suffers from a single point of failure problem.
- **Compatibility with IP Aliasing configurations:** Since the digital certificates and tickets are used to validate ARP replies, the hosts can easily detect the genuine and fake bindings. Thereafter, the hosts can update their primary ARP cache accordingly. Thus, the scheme is compatible with IP Aliasing configuration.

5.6.2 Kernel based Patches

Several kernel based patches are also proposed in the literature as mitigation scheme to prevent ARP poisoning attacks. The Antidote [15] approach is the most popular among them at an individual host level. According to Antidote scheme, when a host receives an ARP reply whose MAC address differs from the previously cached one, it tries to check if the previously learnt MAC address is still alive. If the previously learnt MAC address is still alive, the updated binding is rejected and the offending MAC address is appended to a list of banned addresses.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since this scheme involves probing process to check if the previously learnt MAC address is still alive, it is vulnerable to flooding of spoofed ARP messages.
- **Resistant to IP Exhaustion problem:** There may arise a situation when attacker sends spoofed ARP replies with some random source MAC address. S/he may do so for all the unused IP addresses within the network. After this, whenever the attacker will receive ARP requests for these fake bindings, s/he may send a fake reply so that the detecting host would consider that the older MAC is still alive. The detecting host will now consider that all the IP addresses of the pool are currently in use. Thus, the scheme does not prevent IP Exhaustion problem.
- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.
- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.
- **Compatibility with IP Aliasing configurations:** When this scheme receives an ARP message, it checks if the source IP address of the ARP message is already present in the cache. If it is present, the scheme will check if the previously learnt MAC address is alive. Otherwise, the scheme will simply store the mapping into the cache independent of the source MAC address of ARP message. In the case of IP Aliasing also, until and unless the source IP address of the ARP message is not present in ARP cache, the scheme will continue to learn the IP - MAC mappings. So, it is compatible with IP Aliasing configuration.

5.6.3 Passive Detection

ARPWATCH [16] is one of the most popular tools which works as a passive detection tool. It sniffs the ARP Requests/Replies from the network and constructs a

MAC - IP address mapping database. If it notices a change in any of these mappings in future ARP traffic, the alarm is raised concluding that an ARP spoofing attack is going on.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** This scheme is totally dependent on at what time the attack is launched. If attacker launched the flood of ARP spoofing attack before the detection tool was started for the first time, the tool will learn the spoofed IP - MAC bindings and thus, fake bindings will be stored in address mapping database. However, if the detection tool started its execution before the flood was launched, the flood will be detected easily. Since there is a scenario in which attack cannot be detected, we consider that the scheme is vulnerable to flood of spoofed ARP messages.

- **Resistant to IP Exhaustion problem:** In a network, an attacker can always find the list of IP addresses which is not being used by any host at a particular instant. After this, the attacker can send various spoofed ARP replies (with different randomly generated MAC addresses) on the behalf of unused IP address. As a consequence, the detection tool will store these mappings in the primary ARP cache. Thus, the host, on which detection tool is running, will consider that all the IP addresses are currently in use. Due to this reason, the new incoming hosts in the network will be considered as illegitimate hosts which will finally result into false alarm execution. Thus the scheme is not effective against IP Exhaustion problem.

- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.

- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.

- **Compatibility with IP Aliasing configurations:** Since the detection tool raises the alarm if it notices a change in IP - MAC mappings, this scheme

will create false alarms if the IP Aliasing is configured for some of the MAC addresses. So, this scheme is not compatible with IP Aliasing configuration.

5.6.4 Centralized Detection and Validation Server

Sumit Kumar and Shashikala Tapaswi [17] proposed a centralized technique for detection and prevention of ARP poisoning. In this scheme, an ARP Central Server (ACS) validates the ARP tables' entries of all the hosts within the network. Clients also maintain a secondary long term cache in this scheme.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since the ACS server is based on the probing process to check if the previous IP - MAC mapping is still valid, this scheme is also vulnerable to flood of spoofed ARP messages.
- **Resistant to IP Exhaustion problem:** If an IP - MAC mapping is not present in ACS cache, the attacker can send spoofed ARP messsage with some random MAC address. As a result, the ACS will store this mapping into the cache as well as secondary ARP table. Since, ACS itself contains this mapping, all other hosts within the LAN will honour this mapping. Attacker can send multiple such ARP messages spoofed with different IP address. This will lead to IP Exhaustion problem.
- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.
- **Single point of failure problem:** ACS server, being centralized in nature, creates single point of failure problem.
- **Compatibility with IP Aliasing configurations:** Since the scheme allows mapping of a MAC address with more than one IP address, the scheme is compatible with IP Aliasing configurations.

5.6.5 Usage of ICMP packets as probe packets

Poonam Pandey [19] proposed an approach which involves the usage of ICMP packets as probe packets to validate the ARP messages.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since this scheme involves probing process to validate the ARP messages, it is vulnerable to flooding of spoofed ARP messages.

- **Resistant to IP Exhaustion problem:** By modifying his/her network stack, the attacker can generate corresponding spoofed ICMP replies in response to the probe ICMP echo requests. Using this technique, an attacker can create IP Exhaustion problem.

  There are various tools available which can be used to modify the whole network stack of a computer system. The most popular technique is to use NFQUEUE [27] along with iptables [28]. Though this technique is highly advanced, these tools make it much easier to implement.

- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.

- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.

- **Compatibility with IP Aliasing configurations:** This scheme is compatible with IP Aliasing configurations since the interfaces on which IP aliasing is configured, will reply to all the probe ICMP echo requests destined to them. These IP - MAC bindings will simply be stored in the primary ARP cache.

5.6.6 Using Host based Discrete Event System

Ferdous A. Barbhuiya et al. [20] proposed one scheme using host based Discrete Event System (DES). The scheme is based on a DES model for the system under

normal condition and also under each of the failure conditions. Along with that, a state estimator called diagnoser (or detector, if only detection of failure is required) is designed which observes events generated by the system to decide whether the states through which the system traverses correspond to the normal or faulty DES model. This scheme uses ARP packets as probe packets to validate the ARP replies.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since this scheme involves probing process to validate the ARP messages, it is vulnerable to flooding of spoofed ARP messages.
- **Resistant to IP Exhaustion problem:** An attacker can easily send spoofed ARP Replies saying that different IP addresses (which are not yet verified by the victim host) belong to his/her MAC address. When victim will send probe ARP Requests for these replies to verify them, attacker can send spoofed replies again. As a result, the victim host will store these bindings into the verification table. So, the new upcoming hosts ARP Requests/Replies will be added to the victim host's spoofed table directly though these new hosts are genuine ones. This leads to IP Exhaustion problem.
- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.
- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.
- **Compatibility with IP Aliasing configurations:** Since all the interfaces on which IP Aliasing is configured, will respond to the probe ARP requests, these mappings will simply be stored in primary ARP cache. Thus, the scheme is compatible with IP Aliasing configurations.

5.6.7 ICMP based secondary cache approach

In this approach, we proposed an ICMP based secondary cache [21] to detect and prevent ARP Poisoning. It was shown how the entering and existing algorithms

could detect and prevent these attacks by validating the ARP messages using entries present in secondary ARP table. This scheme uses ICMP echo requests for probing process to check if the previous IP - MAC mapping is still valid.

The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since this scheme involves probing process to validate the ARP messages, it is vulnerable to flooding of spoofed ARP messages.
- **Resistant to IP Exhaustion problem:** Since this scheme allows only one mapping for a MAC address in secondary ARP table, it is resistant to IP Exhaustion attack.
- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.
- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.
- **Compatibility with IP Aliasing configurations:** Since this scheme does not allow more than one mapping for a MAC address in secondary ARP table, it is not compatible with IP Aliasing configurations.

5.6.8 ARP - ICMP probe packet based approach

This is the final scheme proposed by us. We have already discussed the complete working of this solution. The behavior of the scheme based on the five parameters is as follows:

- **Resistant to flood of spoofed ARP messages:** Since both the algorithms (i.e. new host entering algorithm and existing host algorithm) contains the functionality of flood detection, this scheme can easily detect the flood of spoofed ARP messages.

- **Resistant to IP Exhaustion problem:** Since this scheme allows only one mapping for a MAC address in secondary ARP table, it is resistant to IP Exhaustion attack.
- **Backward compatibility with the existing network infrastructure:** Since the scheme does not require any modification in ARP specification, it is backward compatible with existing network infrastructure.
- **Single point of failure problem:** Being distributed in nature, the scheme does not create single point of failure problem.
- **Compatibility with IP Aliasing configurations:** Since this scheme allows network administrators to implicitly define the IP - MAC mappings for IP Aliasing enabled interfaces, it is compatible with IP Aliasing configurations.

Table 5.1 shows a summarized comparison among these schemes based on the five parameters. The checkmark (✓) shows that the scheme's behaviour possesses that property while the crossmark (*X*) shows that the scheme does not possess that property.

Table 5.1: Comparison of various proposed schemes based on the five comparative parameters

	Resistant to flood of spoofed ARP messages	Resistant to IP Exhaustion problem	Backward compatibility with the existing network infrastructure	Resistant to single point of failure problem	Compatibility with IP Aliasing configurations
Cryptography based schemes	✓	✓	*X*	*X*	✓
Kernel based Patches	*X*	*X*	✓	✓	✓
Passive Detection	*X*	*X*	✓	✓	*X*
Centralized Detection and Validation Server	*X*	*X*	✓	*X*	✓
Usage of ICMP packets as probe packets	*X*	*X*	✓	✓	✓
Using Host based Discrete Event System	*X*	*X*	✓	✓	✓
ICMP based secondary cache approach	*X*	✓	✓	✓	*X*
Our proposed method	✓	✓	✓	✓	✓

CHAPTER 6
CONCLUSION

We proposed a decentralized scheme that can be a possible solution to detect and prevent ARP Poisoning and IP Exhaustion attacks. The proposed algorithms validate and store the MAC - IP mappings into secondary ARP table. IP Aliasing table allows network administrators to specify the MAC - IP mappings for interfaces configured with IP aliasing. An alarm module is also added with the proposed scheme. The scheme uses ARP and ICMP protocols to generate probe requests. Secondary table does not allow multiple entries for a single IP address or MAC address so as to prevent the attacks.

The scheme does not require any modifications in the existing ARP specifications. All the components included in the scheme are backward compatible with the existing networks. Also, the scheme can be used even when IP aliasing is configured for some of the interfaces. Being a decentralized scheme, it does not create a single point of failure problem. A flood of spoofed ARP Replies can also be detected using this scheme. The scheme's probing process prevents it from limitations which were present in passive detection techniques.

We presented different attack scenarios theoretically and have shown that the scheme is effective under all possible scenarios. Also, we presented the detection and prevention of classical ARP poisoning and IP Exhaustion attacks using experimental results. The generated traffic is analyzed to show the efficiency of scheme. A comparative analysis is also presented between our proposed scheme and other schemes based on the five parameters.

The direction of future work includes the development of an efficient mechanism using which the secondary ARP table can be distributed among the hosts. It will reduce the learning time of all the hosts.

LIST OF AUTHOR'S PUBLICATIONS

1. Nikhil Tripathi, B. M. Mehtre. "An ICMP based Secondary Cache approach for the detection and prevention of ARP poisoning", 4^{th} *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, December 26th - 28th, pp: 1-6, (2013)

   Available at: `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6724172&abstractAccess=no&userType=inst`

2. Nikhil Tripathi, B. M. Mehtre. "Analysis of various ARP Poisoning mitigation techniques : A comparison", *(Accepted in IEEE International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT 2014))*.

3. Nikhil Tripathi, B. M. Mehtre. "DoS and DDoS Attacks: Impact, Analysis and Countermeasures", *TEQIP II National Conference on Advances in Computing, Networking and Security (NCACNS)*, Nanded, India, December 23rd - 24th, pp. 93-98, (2013)

PUBLICATIONS COMMUNICATED

1. Nikhil Tripathi, B. M. Mehtre. "An ARP - ICMP probe packet based scheme to prevent ARP Poisoning and IP Exhaustion Attacks", *(Communicated to International Journal of Information Security (IJIS), Springer).*

BIBLIOGRAPHY

[1] Cappelli, Dawn, Andrew Moore, Randall Trzeciak, and Timothy J. Shimeall. ”Common Sense Guide to Prevention and Detection of Insider Threats.” CERT, Jan. 2009. Web. 25 May. 2014.

[2] Fyffe, George. *Addressing the Insider Threat*. Network Security. Mar. 2008: Science Direct. Web. 5 June. 2014.

[3] Jennings, Frank. *Beware the Enemy Within*. SC Magazine. Jul. 2008: Business Source Complete. Web. 5 June. 2014.

[4] Kaplan, D.. *Internal Review*. SC Magazine. 1 Feb. 2011: ABI/INFORM Trade & Industry, ProQuest. Web. 5 June. 2014.

[5] Blades, M.. *The Insider Threat*. Security Technology Executive. 1 Nov. 2010: ABI/INFORM Trade & Industry, ProQuest. Web. 5 June. 2014.

[6] Nikhil Tripathi, BM Mehtre, *DoS and DDoS Attacks: Impact, Analysis and Countermeasures* in Proceedings of National Conference on Advances in Computing, Networking and Security (NCACNS), Nanded, India, December 23rd - 24th, pp: 93-98, (2013)

[7] Maxi, Merritt. ”Defending against Insider Threats to Reduce Your IT Risk.” Security and Compliance, Jan. 2011. Web. 25 May 2014.

[8] ”Cyber-Ark; Cyber-Ark Global Survey Shows External Cyber-Security Risks Will Surpass Insider Threats. ” Investment Weekly News. 30 Apr. 2011: ABI/INFORM Trade & Industry, ProQuest. Web. 25 May. 2014.

[9] David C. Plummer, *An Ethernet Address Resolution Protocol*, RFC 826, Internet Engineering Task Force, November 1982.

[10] Kozierok, C.M., *TCP/IP Guide*. first ed. No Starch Press. 2005. San Francisco. 5 February. 2014

[11] Hou, X., Jiang, Z., Tian, X.,: The detection and prevention for ARP spoofing based on snort. In: IEEE International Conference on Computer Application and System Modeling (ICCASM), Taiyuan, October 22nd - 24th, pp. 137 - 139, (2010)

[12] Bruschi, D., Ornaghi, A., Rosti, E.,: S-ARP: a secure address resolution protocol. In: Proceedings 19th IEEE Annual Conference on Computer Security Applications (CSAC), pp. 66 - 74, (2003)

[13] Lootah, W., Enck, W., McDaniel, P.,: TARP: Ticket-based address resolution protocol. In: 21st IEEE Annual Conference on Computer Security Applications (CSAC), pp. 106 - 116, (2005)

[14] Nam, S., Kim, D., Kim J.,: Enhanced ARP: preventing ARP poisoning based man-in-the-middle attacks. In: IEEE Communications Letters, vol. 14 , pp. 187 - 189, (2010)

[15] Teterin, I.,: *Antidote*. (2003) Available at: `http://antidote.sourceforge.net`

[16] Leres, C.,: *ARPWATCH tool: ARP Spoofing Detector*. (2006) Available at: `ftp://ftp.ee.lbl.gov/arpwatch.tar.gz`

[17] Kumar, S., Tapaswi, S.,: A centralized detection and prevention technique against ARP poisoning. In: IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Kuala Lumpur, Malaysia, June 26th - 28th, pp. 259 - 264, (2012)

[18] Jinhua, G., Kejian, X.,: ARP spoofing detection algorithm using ICMP protocol. In: IEEE International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, September 11th - 13th, pp. 1 - 6, (2013)

[19] Pandey, P.,: Prevention of ARP spoofing: A probe packet based technique. In: IEEE International Advance Computing Conference (IACC), Ghaziabad, India, February 22nd - 23rd, pp. 147 - 153, (2013)

[20] Barbhuiya, F. A., Biswas, S., Hubballi, N., Nandi, S.,: A host based DES approach for detecting ARP spoofing. In: IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Paris, France, April 11st - 15th, pp. 114 - 121, (2011)

[21] Tripathi, N., Mehtre, B. M.,: An ICMP based secondary cache approach for the detection and prevention of ARP poisoning. In: IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Enathi, India, December 26th - 28th, pp. 1 - 6, (2013)

[22] Biondi, P.,: *Scapy*. (2011) Available at: `http://www.secdev.org/projects/scapy/`

[23] Black, A. W.,: *Festival*. (2010) Available at: `http://www.cstr.ed.ac.uk/projects/festival/`

[24] Hipp, M., Fromme, O.,: *mpg123*. (1999) Available at: `http://www.mpg123.de/`

[25] Aharoni M., Kearns, D.,: *Kali Linux*. (2012) Available at: `http://www.kali.org/`

[26] The Wireshark Team,: *Wireshark*. (2013) Available at: `http://www.wireshark.org/`

[27] *packEth tool*. Available at: http://packeth.sourceforge.net/packeth/Home.html

[28] *NFQUEUE tool*. Available at: http://www.ohloh.net/p/nfqueue-bindings

[29] *iptables tool*. Available at: http://www.netfilter.org/projects/iptables/index.html