A Review on ARP Spoofing Detection and Prevention

Francis Jaison[1], Amitha Joseph[2]

[1]UG Scholar, Santhigiri College of Computer Sciences Vazhithala,
francisjaison78@gmail.com

[2]Asst. Prof. Santhigiri College of Computer Sciences Vazhithala,
amithajoseph@santhigiricollege.com

Abstract: ***Address Resolution Protocol (ARP) may be a procedure for mapping a dynamic net Protocol address (IP address) to a permanent physical machine address during a native space network (LAN). The physical machine address is additionally called a Media Access management or mackintosh address. All operational systems in associate IPv4 LAN network keep associate Arp cache. Each time a number requests a mackintosh address so as to send a packet to a different host within the computer network, it checks its Arp cache to examine if the scientific discipline to mackintosh address translation already exists. If it does, then a replacement Arp request makes no sense. Arp broadcasts a call for participation packet to all or any the machines on the computer network and asks if any of the machines understand they're victimization that exact scientific discipline address. Once a machine acknowledges the scientific discipline address as its own, it sends a reply thus Arp will update the cache for future reference and proceed with the communication. Arp spoofing may be a variety of attack during which a malicious actor sends falsified Arp (Address Resolution Protocol) messages over a local area network. This ends up in the linking of associate attacker's mackintosh address with the scientific discipline address of a legitimate laptop or server on the network. Once the attacker's mackintosh address is connected to associate authentic scientific discipline address, the assailant can begin receiving any information that's meant for that scientific discipline address. Arp spoofing will modify malicious parties to intercept, modify or perhaps stop information in-transit. Arp spoofing attacks will solely occur on native space networks that utilize the Address Resolution Protocol.***

Keywords: ARP spoofing, ARP cache poisoning, ARP poison routing, denial of service, man in the middle attack

1. Introduction

The Arp protocol is one amongst the foremost basic however essential protocols for local area network communication. The Arp protocol is employed to resolve the mackintosh address of a bunch given its information processing address. This is often done by causation associate Arp request packet (broadcasted) on the network. The involved host currently replies back with its mackintosh address in associate Arp reply packet (unicast). In some things a bunch would possibly broadcast its own mackintosh address during a special Gratuitous Arp packet. All hosts maintain associate Arp cache wherever all address mappings learnt from the network (dynamic entries) or configured by the administrator (static entries) area unit unbroken. The dynamic entries age out once a fixed interval of your time that varies across operational systems. Once the entry ages out it's deleted from the cache and if the host desires to speak with identical peer, another Arp request is formed. The static entries never age out. The Arp protocol is homeless. Hosts can cache all Arp replies sent to them notwithstanding they'd not sent a definite Arp request for it. Notwithstanding a previous valid dynamic Arp entry is there within the Arp cache it'll be overwritten by a more modern Arp reply packet on most operational systems. All hosts blindly cache the Arp replies they receive, as they need no mechanism to evidence their peer. This is often the foundation drawback that results in. Arp is that the method of shaping Arp packets to be ready to impersonate another host on the network. Within the most general kind of the assaulter sends spoofed Arp responses to the victim sporadically. Amount between the spoofed responses is way lesser than the Arp cache entry timeout period for the software package running on the victim host. This may make sure that the victim host would never build associate Arp request for the host whose address the assaulter is impersonating. Following area unit some sorts of attacks that may be resulted from Arp Spoofing: Man-in-the-Middle (MIM), Denial of Services (DoS). Man-in-the-Middle (MIM) may be a quite common form of attack, within which associate assaulter inserts his pc between the communication methods of 2 target computers by Sniffs packets from Network, changed them and so insert them back to the Network. The malicious pc can forward frames between the 2 computers; therefore communications don't seem to be interrupted, however all traffic 1st goes to the assaultive pc instead of targeting and victim computers. A “Denial of Service (DoS)” attack may be a flood of packets that consumes network resources and causes impasse. Through "Denial-of-Service (DoS)" attack attackers build the system unusable and stop services from victimization for legitimate user by overloading, damaging or destroying resources so the services can't be used. DoS Attack is performed with associate assaulter sterilization the hosts Arp Cache (by Arp Poisoning) with non-existent entries (MAC Addresses). This cause frames to be born due to the restricted size of Arp Cache.

2. Literature Survey

2.1. ARP

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

Working of ARP Messages is shown in Fig. 1

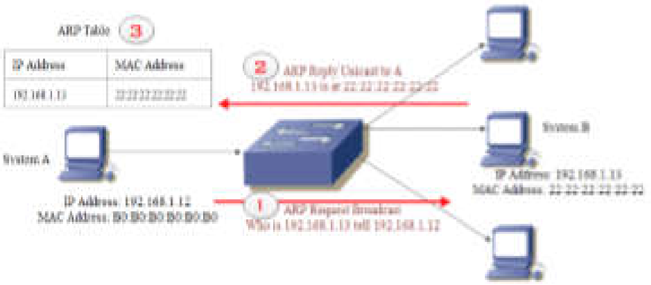

Fig. 1 ARP request by system A and ARP reply from system B

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

2.2 ARP Cache Poisoning

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

```
#! /bin/bash echo1 /proc/sys/net/ipv4/ip forward arpspoof -i eth0 -t 192.168.1.32 192.168.1.1 & arpspoof -i eth0 -t 192.168.1.1 192.168.1.32 &
```

When Victim broadcast an ARP Request for gateway. The Attacker replies with ARP Reply packets and effectively poison the victims ARP Cache. Thus, the attacker becomes MITM between gateway and victim by: 1) poison the victim so that gateways IP address gets mapped with attackers MAC address and 2) poison the gateway so that victims IP address gets associated with attackers MAC address and 3) forwarding the packets the attacker receives to victim/gateway. Now the attacker is MITM between victim and gateway.

3. Methods/Approach

3.1. Secure ARP Protocol (S-ARP)

The S-ARP protocol is definitely a permanent solution to ARP spoofing but the biggest drawback is that we will have to make changes to the network stack of all the hosts. This is not very scalable as going for a stack upgrade across all available operating systems is something both vendors and customers will not be happy about. As S-ARP uses Digital Signature Algorithm (DSA) we have the additional overhead of cryptographic calculations.

3.2. Static MAC Entries

Adding static mackintosh addresses on each host for all alternative hosts won't however it isn't a scalable resolution in any respect and managing of these entries may be a full time job by itself. This could fail miserably if mobile hosts like laptops square measure sporadically introduced into the network. Additionally some in operation systems square measure known to write static Arp entries if they receive Gratuitous Arp packets (GARP).

3.3. Kernel Based Patches

Kernel based mostly patches like Anticap and Antidote have created a shot to guard from Arp spoofing at an individual host level. Anticap doesn't enable change of the host Arp cache by Associate in Arp reply that carries a different mackintosh address then the one already within the cache. This sadly makes it drop legal gratuitous Arp replies moreover, that may be a violation to the Arp protocol specification. Remedy on receiving Associate in Arp reply whose mackintosh address differs from the antecedently cached one tries to see if the antecedently learnt mackintosh continues to be alive. If the antecedently learnt mackintosh continues to be alive then the update is rejected and also the and also the address is additional to an inventory of illegal addresses. Each the higher than techniques accept the actual fact that the Arp entry within the cache is that the legitimate one. This creates a race scenario between the assaulter and also the victim. If the assaulter gets his spoofed Arp entry into the hosts' cache before the important host will, then the important mackintosh address is illegal. This may solely be undone by body intervention. Therefore we will conclude that wrong learning could cause these tools to fail in detection of Arp spoofing.

3.4. Network Analyzer Tools and Sniffers

Network Analyzer Tools and Sniffers are most useful tools available on Internet that can be used to debugging network problems.

They are used by network professionals to diagnose network abnormalities. It allows you to inspect network traffic at every level of the network stack in various degrees of detail. ARP Watch, Dsniff, Hunt, Parasite are some popular tools.

3.5. Encryption

Encryption is an efficient thanks to defend against Sniffing and Arp Spoofing. Secret writing prevents any non-authorized party from reading or dynamical information.

The level of protection provided by secret writing is set by associate degree secret writing algorithmic program. During a powerful attack, the strength is measured by the amount of doable keys and therefore the key size. If communication between hosts systems is encrypted at the Network Layer there's very little likelihood for programs like Dsniff to collect helpful info from the Network, since the assaulter won't understand that packets contains authentication info and that don't.

3.6. Intrusion Detection Systems (IDS)

IDS determine attacker's tries to attack or entered the network and misuse it. IDSs might monitor packets passing over the network, monitor system files, monitor log files, or discovered deception systems that commit to lure hackers.

Port Scans and Denial-of-Service Attacks area unit an in progress threat. Intrusion Detection System is vital parts of a defense-in-depth security resolution that may determine potential threats and permit you to require immediate action to dam a hacker or a selected IP address that is being employed to launch an assault.

There are a two varieties of IDSs available: Network-Based IDSs (NIDS) and Host-Based IDs (HIDS).

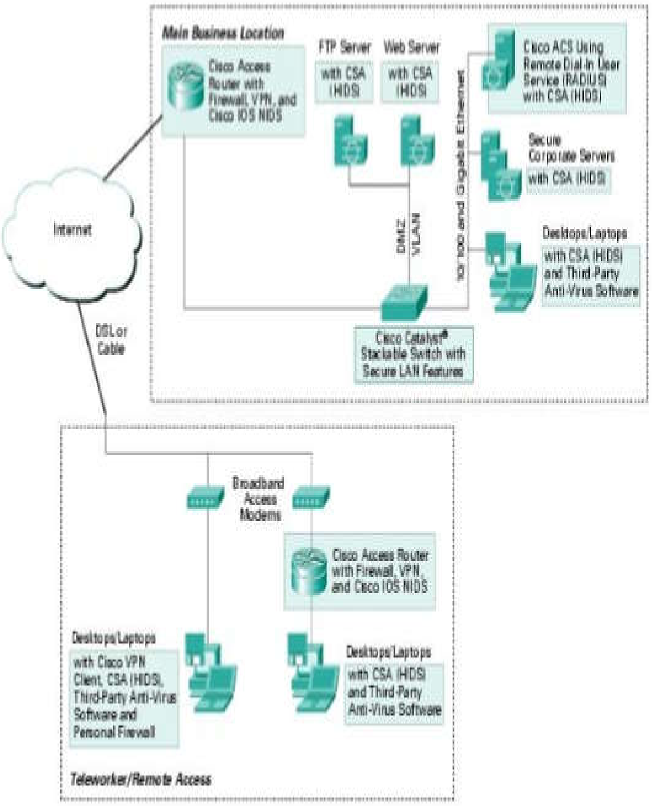
Main Business Location
Cisco Access Router with Firewall, VPN, and Cisco IOS NIDS
FTP Server with CSA (HIDS)
Web Server with CSA (HIDS)
Cisco ACS Using Remote Dial-In User Service (RADIUS) with CSA (HIDS)
Secure Corporate Servers with CSA (HIDS)
Desktops/Laptops with CSA (HIDS) and Third-Party Anti-Virus Software
10/100 and Gigabit Ethernet
DMZ VLAN
Cisco Catalyst® Stackable Switch with Secure LAN Features
Internet
DSL or Cable
Broadband Access Modems
Cisco Access Router with Firewall, VPN, and Cisco IOS NIDS
Desktops/Laptops with Cisco VPN Client, CSA (HIDS), Third-Party Anti-Virus Software and Personal Firewall
Desktops/Laptops with CSA (HIDS) and Third-Party Anti-Virus Software
Teleworker/Remote Access

Fig.2 An Example of Implementing IDS in the Network

3.7. Passive Detection

In Passive Detection we tend to the Arp requests/responses on the network and construct a mackintosh address to informatics address mapping info. If we tend to notice a modification in associate degree of those mappings in future Arp then we tend to raise associate degree alarm and conclude that an Arp attack is afoot. The foremost common tool during this class is ARPWATCH. The most downside of the passive methodology may be a break between learning the address mappings and consequent attack detection. In a very scenario wherever the Arp traffic began before the detection tool was started for the first time, the tool can learn the cast replies in its informatics to mackintosh address mapping info. Currently solely once the victim starts human activity with another host the inconsistency are detected associate degreed an alarm raised. The wrongdoer might have created his getaway owing to this delay. Conjointly a spoofed entry learned as within the higher than situation would have to be compelled to be manually undone by the network administrator. The sole resolution to the current drawback is to manually feed the proper address mappings into the info before beginning the tool or produce associate degree attack free learning traffic. Each of those square measure unreasonable because of measurability and quality problems. A perfect example would be mobile hosts e.g. laptops brought in by customers or guests to an organization. This slow learning curve makes it not possible to put in passive tools on an outsized network (1000+ hosts) and expect them to spot attacks outright. The passive techniques don't have any intelligence and blindly seek for a match within their learnt info tables. If associate degree Arp traffic is detected than there's no manner of ascertaining if the recently seen address mapping is owing to try or the antecedently learnt one was truly a spoofed one. Our technique can confirm the important mackintosh to informatics mapping throughout associate degree actual attack to a good degree of accuracy. The passive learning technique is additionally terribly unreliable. A brand new address mapping is learnt once Arp is seen from them. Therefore a switch Arp Cache table overflow try by the generation of random Arp reply packets per second with discretionary mackintosh and informatics addresses can simply lead to new stations being discovered rather than being according as attack traffic.

3.8 The Proposed Active Detection Technique for ARP spoofing

The proposed technique actively interacts with the network to gauge the presence of ARP spoofing attacks. We will henceforth assume the following about the network we desire to protect.

3.8.1 Assumptions

1. The attacker’s pc includes a traditional network stack. This assumption can hold for many of the attacks as “ready to use” creative person tools have perpetually been the attacker’s preferred alternative. If the assaulter will use a custom stack then our technique can still discover creative person however won't be ready to predict the right address mappings any longer.
2. The individual hosts we tend to need to guard on the network could use a private firewall and a minimum of one TCP port ought to be allowed through the firewall. This is often to permit our probe packets (TCP SYN packets) to travel through. this is often an inexpensive assumption as although is put in some computer network primarily based services like NETBIOS etc. area unit usually allowed through it for computer network communication. We assume that all devices, which we protect, have a TCP/IP network stack up and running.

3.8.2. Terminology

We now introduce the terminology used in the rest of this paper.

1. Threshold interval: Arp replies to associate Arp request should be received inside a specified measure. Once this point has elapsed we are going

to contemplate the Arp request to possess "expired". We are going to decision this interval because the "Threshold Interval". This may be administratively configurable on any tool victimization our technique.

2. Host Database: This can be the mapping of all legitimate IP-mackintosh pairs on the network verified and learnt by our technique.

The ARP packets consist of the MAC header and the ARP header. Based on the value of the source and destination MAC addresses in the MAC header and as advertised in the ARP header we can divide the all ARP packets into 2 categories.

1. Inconsistent Header ARP packets: The MAC addresses in the MAC and ARP header differ i.e. Source MAC address in MAC header! = Source MAC address in ARP header (in ARP requests/responses) and/or Destination MAC address in MAC header! = Destination address in ARP header (only for ARP replies).
2. Consistent Header ARP packets: These are the compliment of the Inconsistent Header ARP packets. The MAC addresses in the MAC and ARP headers match in these packets.

Note that Inconsistent Header ARP packets are guaranteed spoofed packets, as such an anomaly is only possible in attack traffic. Based on the above classification we can further bunch the Consistent Header ARP packets into three groups:

1. Full ARP Cycle: An ARP request and its corresponding ARP replies seen within the threshold interval.
2. Request Half Cycle: An ARP request for which no replies are sent as seen within the threshold time.
3. Response Half Cycle: An ARP reply generated without an ARP request.

These three categories form the basis of our input to the ARP spoofing detection mechanism. The following subsection discusses the Architecture of the proposed technique in detail.

3.8.3 Architecture

Please refer to Figure 1 for the architecture discussion. We have adopted a modularized approach and have divided our spoof detection into the following modules:

1. ARP Sniffer module: This sniffs all ARP traffic from the network.
2. MAC - ARP header anomaly detector module: This module classifies the ARP traffic into Inconsistent Header ARP packets and Consistent Header ARP packets.
3. Known Traffic Filter module: This filters all the traffic, which is already learnt. It will either drop the packet if the IP to MAC mapping is coherent with the learnt Host Database or raise an alarm if there are any contradictions. All the new ARP packets with unknown addresses are sent to the Spoof Detection Engine for verification.
4. Spoof Detection Engine module: This is the main detection engine. We feed the Consistent Header ARP packets to it as input. The design of this module will be discussed in Section 3.8.4.
5. Add to Database Module: Legitimate ARP entries verified by the Spoof Detection Engine are added to the Host Database by this module.
6. Spoof Alarm Module: This module raises an alarm on detection of ARP spoofing by sending a mail, SMS etc. to the administrator.

As shown in Figure 3, the Arp module sniffs all the Arp traffic in its computer network section and passes it to the mackintosh – Arp Header Anomaly Detector. This module passes the whole Consistent Header Arp packets to the famous Filter module. The whole Inconsistent Header Arp packets square measure sent to the Spoof Alarm. This can be done as a result of the Inconsistent Header Arp packets square measure all spoofed packets as mentioned earlier. The famous Filter module can take away all traffic coherent with the already learnt addresses by consulting the Host information. If there's a contradiction with the already learnt addresses then it raises a Spoof Alarm. All new Arp traffic is passed to the Spoof Detection Engine.

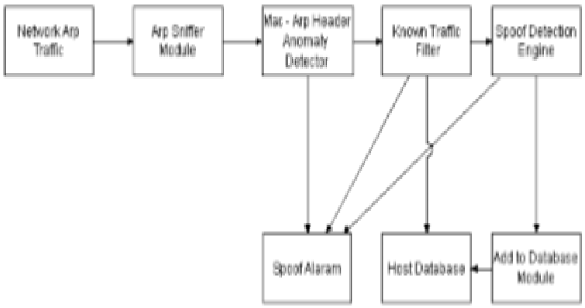

Fig. 3. Inter-relation between various Modules used by the ARP Spoof Detection Algorithm

The Spoof Detection Engine applies our observation algorithmic program to detect Arp spoofing. The new seen Consistent Header Arp packets are input to the current module. The engine currently internally bunches these packets into the 3 classes mentioned in Section 3.8.4 specifically Full Arp Cycle, Request and Response Half Cycle packets. The detection algorithmic program applied by the engine are mentioned within the section 3.8.5. Once applying the detection algorithmic program the Spoof Detection engine either sends the Arp entry to the boost info module or the Spoof Alarm module. The boost info module can add these verified waterproof and science address mapping to the Host info. The spoof detection engine is mentioned thoroughly next

3.8.4 The Spoof Detection Engine

The Spoof Detection Engine is that the heart of the complete system. The 3 Arp Cycle packets as mentioned in Section 3.8.2 are treated in slightly different ways in which by the Spoof Detection Engine to sight attempt to spoof. The Spoof Detection Engine works supported the subsequent Rules:

Rule A: "The network interface card of a bunch can settle for packets sent to its mackintosh address, Broadcast address and signed multicast addresses. It'll depart this world these packets to the internet protocol layer. The internet protocol layer can solely settle for internet protocol packets addressed to its internet protocol address(s) and can taciturnly discard the remainder of the packets. If the accepted packet may be a protocol packet it's passed on to the protocol layer. If a protocol SYN packet is received then the host can either respond back with a protocol SYN/ACK packet if the destination port is open or with a protocol RST packet if the port is closed".

Rule B: "The offender will spoof Arp packets impersonating a bunch however he will ne'er stop the real host from replying to Jean Arp requests (or the other packet) sent thereto. The valid assumption here is that the real host is informed the network."

It ought to be noted that these rules are derived from the right behavior that a host's network stack ought to exhibit once it receives a packet. To exemplify Rule A, let a bunch have mackintosh address = X and internet protocol address = Y. If this host receives a packet with destination mackintosh address = X and destination internet protocol address = Z then despite the fact that the network interface card would settle for the packet because the destination mackintosh address matches, the host's network stack can taciturnly discard this packet because the destination internet protocol address doesn't match, while not causing any error messages back to the supply of the packet.

It ought to be noted that these rules are derived from the proper behavior that a host's network stack ought to exhibit once it receives a packet. To exemplify Rule A, let a bunch have mackintosh address = X and internet protocol address = Y. If this host receives a packet with destination mackintosh address = X and destination internet protocol address = Z then even supposing the network interface card would settle for the packet because the destination mackintosh address matches, the host's network stack can taciturnly discard this packet because the destination internet address doesn't match, while not causation any error messages back to the supply of the packet.

Based on Rule A, we are able to ideate 2 styles of probe packets from a host's network stack purpose of read that we'll use to discover Arp spoofing. a. Right mackintosh – Wrong IP packet: The destination mackintosh address within the packet is of the host however the IP address is invalid and doesn't correspond to any of the host's addresses. The destination host can wordlessly drop this packet. b. Right mackintosh – Right IP packet: The destination mackintosh address and IP addresses pairs are of the host's and its network stack accepts it.

We will henceforward assume that the assaulter is mistreatment an associate network stack. The performance of our technique within the presence of a modified network stack are evaluated in Section 3.8.5. Supported the higher than observation we are going to construct our own packets supported Rule A and send them on the network. We are going to use the address data within the Arp response packet sent by the host whose believability is to be verified. we are going to use the mackintosh and internet protocol addresses employed in the Arp response packet to construct a protocol SYN packet i.e. the destination mackintosh and internet protocol within the protocol SYN packet are the supply mackintosh and internet protocol address advertised within the Arp response packet internet protocol supply mackintosh and internet protocol within the protocol SYN packet would be of the host running the Spoof Detection Engine. The protocol destination port are chosen supported the presence/absence of packet filtering firewalls on the network hosts. If there's put in on the hosts we will going to select the "allowed protocol port" and if no firewalls are there then we are able to select any protocol port. The remainder of the header values within the protocol SYN packet are set as was common.

When a TCP SYN packet as constructed above is sent to the source of the ARP reply packet, the host's response will be based on Rule A. If the ARP response was from the real host its IP stack will respond back with either a TCP RST packet (If the destination port is closed) or a TCP SYN/ACK packet (if the destination port is open).

If the ARP response had been from a malicious host then its network stack would silently discard the TCP SYN packet in accordance with Rule A. Thus based on the fact that the Spoof Detection Engine does/does not receive any TCP packets in return to the SYN packet it sent, it can judge the authenticity of the received ARP response packet.

We will now discuss how Rules A and B can be used together to detect ARP spoofing attempts in a network. Please refer to Figure 4 for a diagrammatic representation of the algorithm in the form of a flow chart. As we had mentioned earlier the ARP packets are classified into the 3 cycles namely Full ARP Cycle, Request and Response Half Cycles and then fed as input to the Spoof Detection Engine. We will now discuss the application of the above discussed technique to these 3 Cycles to detect ARP spoofing.

3.8.4.1 Full ARP Cycle

A Full ARP Cycle will consist of an ARP request and one or more responses. We will send TCP SYN packet(s) constructed using the MAC and IP address information in the ARP reply packet(s) to the source host(s) as mentioned previously in Section 3.8.4. Based on Rule A only the real host will reply back with either a TCP SYN/ACK or RST packet. We will add this entry into our Host Database as a legitimate MAC to IP address mapping. All other ARP replies which were part of the recorded Full ARP Cycle are spoofed replies and the module will raise a Spoof Alarm for their addresses.

Note that not only we have detected spoofing but also have successfully detected the MAC to IP address mapping of the true host on the network, as only the true host's network stack replies to TCP SYN probes as per Rule A.

3.8.4.2 Request Half Cycle

A Request half cycle might arise when either the destination host is down on the network. If the source IP of the ARP request packet is unknown and not in our Host Database then we will send a TCP SYN packet constructed as mentioned in Section 3.8.4 from the source MAC and IP address information advertised in the ARP request packet. If we get a TCP SYN/ACK or RST packet in response then the host is authentic else we raise a Spoof Alarm. As an alternative way of detecting spoofing we also send an ARP Request packet to the sender of the Request Half Cycle and we will raise a Spoof Alarm if we do not get the same MAC address in the ARP Response packet from the host in return. We will use both these mechanisms simultaneously to detect spoofing. The latter method will come in handy when the attacker uses a customized stack which we will discuss in Section 3.8.5. Figure 3 only contains the TCP method flow for simplicity.

3.8.4.3 Response Half Cycle

A Response Half Cycle could arise because of two situations:

1. It is an ARP spoofing attempt by a malicious attacker. This is one of the most common ways of orchestrating an
   ARP spoofing by sending periodic spoofed ARP response packets to the victims so that the spoofed address entry never expires in the victim's ARP cache.
2. We may have missed the ARP request. This may happen if the detection tool just came online after the ARP request was sent and so we could only sniff the ARP response. Another remote possibility is we missed a packet because of a huge number of packets coming in and inadequate buffer space in the input queue.

To probe the authenticity of the sender of the ARP response we first send an ARP request packet corresponding to the ARP response packet i.e. the destination IP address in the constructed ARP request = the source IP address of the received ARP response and the source IP address of the constructed ARP request = Spoof Detection Engine's host's IP address. The Source MAC address of the constructed ARP request = Spoof Detection Engine's host's MAC address and the destination MAC address will be the broadcast address.
By Rule B even if an attacker is spoofing ARP packets on the network he cannot stop the real host from replying to an ARP request sent to it. As the destination MAC address of an ARP request is the broadcast address so every host will receive it. Thus when we send the above packet out to the network there could be two possible responses:

1. One or more ARP responses: We will now consider our ARP request and these ARP responses as a single Full ARP Cycle. This Full ARP Cycle will now be dealt with as in Section 3.8.4.1 to detect spoofing.
2. No ARP responses: If we do not receive any ARP responses than most probably the real host is down and the ARP responses we see are by an impersonating attacker. To detect this we send a TCP SYN packet constructed as in Section 3.8.4 based on the information in the received ARP response packet. We will find that the impersonating host will not reply to this TCP packet as its network stack will discard it according to Rule A and we will raise a Spoof Alarm.

Thus we have successfully shown how we can detect ARP spoofing attacks in a network using our active injection technique. Till now we have assumed that the attacker is using a normal network stack and orchestrates all these attacks with ready to use tools such as ARP-SK. We will now discuss the performance of our technique in the presence of a customized network stack used by the attacker.

3.8.5 Attacker Uses a Customized Stack

Let us assume that the attacker is aware of our proposed method and has customized his network stack to reply to the TCP SYN packets and ARP request packets destined for the real host, he desires to impersonate. Even in such a scenario we will be able to detect ARP spoofing successfully using Rule B. The only limitation now would be that we would not be able to detect the real MAC and IP address as in the previous case.

Almost all ARP spoofing techniques continuously send spoofed ARP response packets to the victims. This is done so that the victim never needs to raise an ARP request, as the ARP cache entry for the host who's MAC is being spoofed never ages out. But if we send an ARP request on the network, requesting for the MAC address of the host (whose address is being spoofed) the host will reply with an ARP response packet (Rule B). Now we will have a MAC address mismatch for the same IP as the spoofed replies sent by the attacker previously will carry a different MAC address.

We will now discuss our performance for a customized network stack in the light of the ARP Cycles:

1. Full ARP Cycle: If spoofing is on then, both the Attacker and the real host will reply to the original ARP request and we can detect a conflict in the MAC address for the same IP. Also if we send a TCP SYN packet to the source address of both the ARP replies than we will receive two TCP packets in response as the attacker's customized stack replies as well along with the real host. This makes it very easy to detect spoofing.
2. Request Half Cycle: As outlined in Section 3.8.4.2 we will try to authenticate the sender of the request by sending an ARP request packet back to the sender and check the reply(s) for spoofing. If the source MAC addresses in the ARP reply packet received for the injected ARP request does not match the MAC address in the ARP Request Half Cycle packet then we will raise a Spoof Alarm.
3. Reply Half Cycle: If a customized stack is used we will get multiple replies to the ARP request we send as in Section 3.8.4.3. Also when we send out TCP SYN packets to the sources of the ARP request we will get multiple TCP (SYN/ACK or RST) packets in return with different MAC addresses. This is enough to conclude that a spoofing is going on.

Note that though we can detect ARP spoofing even in the presence of an attacker aware of our methods and using a customized stack we cannot predict the correct MAC to IP address mapping. This is the only limitation of our method in the presence of a customized stack.

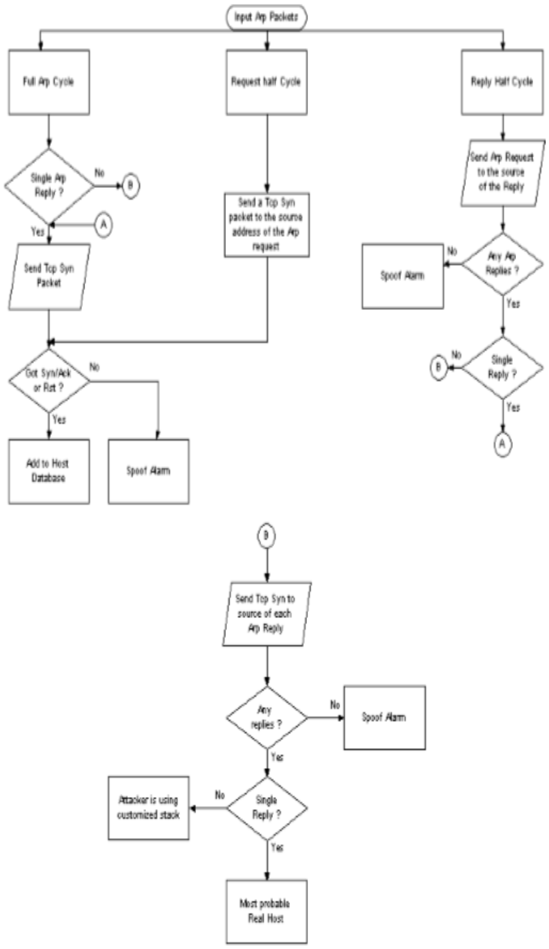

Fig 4. Flow Chart Representation of the Spoof Detection Engine

4. Results

Our technique is clearly much faster and reliable than the passive detection techniques. This technique can be used in a large network and it will immediately detect ARP spoofing attacks even if the attack had begun before the tool using our technique was operational. This is because the time lag between learning and detection is very less as we probe the authenticity of hosts as soon as we see ARP traffic from them. Also our technique verifies the authenticity of the ARP traffic on the network and does not blindly add newly seen traffic to its database. Even in the event of an actual attack our technique can detect the correct IP to MAC address mapping of the real host in the absence of the attacker using a customized network stack. If the attacker uses a customized stack, which replies to our probes we are still able to detect ARP spoofing but will not be able to predict the real MAC to IP address mapping. So even in our worst-case scenario (in the presence of a customized stack) our performance is still better than using a Passive detection technique.

5. Conclusion

Practice security is constantly changing process. ARP is not secure and easy to fool. We need stronger mechanism to enforce security. We must be aware of the fact that switches are not security tools.

Possibility of ARP Spoofing Attacks can be reduced by configuring the network to decline packets from the Internet that claim to originate from the local address. Second thing, proper router configuration in a router is also a good option for security. Most of the Attacks happen because of the Improper Router Configuration. Here one thing is important if the network trusts foreign hosts, routers will not protect against a spoofing attack that claims to originate from those hosts and if you allow internal addresses to access through the outside portion of the firewall, you are vulnerable to Attacks too.

All these problems are caused by the trust-relationship between one host and the other. With the current IP protocol technology, it is quite impossible to eradicate Spoofing. Better way to prevent Spoofing is by using IPv6 or IPsec instead of IPv4, which include two new characteristics authentication header and encapsulated security payload.

Finally, Network Control Mechanisms are dangerous, and must be carefully guarded.

6. References

[1] http://wiki.cas.mcmaster.ca
[2] https://en.wikipedia.org
[3] https://afteracademy.com
[4] https://www.veracode.com
[5] http://www.infosecwriters.com
[6] https://www.researchgate.net

Author Profile

Francis Jaison Pursuing Bachelor of Computer Application from Santhigiri college of Computer Sciences, Vazhithala in 2018-2021

Amitha Joseph received the MCA professional degree and MPhil in Computer Science. She is currently working as an assistant professor in Santhigiri College of Computer Sciences, Vazhithala.