

COMPARATIVE STUDY ON METHODS USED IN PREVENTION AND DETECTION AGAINST ADDRESS RESOLUTION PROTOCOL SPOOFING ATTACK

TSEHAY ADMASSU ASSEGIE¹, PRAMOD SEKHARAN NAIR²

¹Lecturer. Aksum University, Department of Computing Technology, Aksum, Ethiopia

²Professor. Medi-Caps University Indore, Department of Computer Science and Engineering, MP, India

E-mail: ¹tsehayadmassu2006@gmail.com, ²pramodsnair@yahoo.com

Correspondence E-mail: tsehayadmassu2006@gmail.com

ABSTRACT

Address Resolution Protocol spoofing attack is the most common type of local area network attacks. This is because the protocol packet does not contain any authentication information, which indicates the origins of reply packet. Therefore, all of the devices in Local Area Network are vulnerable to this attack. A tool like, ARPspoofer can be used to generate a forged Address Resolution Protocol reply packet to perform the attack, even without any knowledge of the details behind address resolution process. The existence of such automated tools has created a hole for the attackers to easily attack a host in a local area network. And although it is underestimated attack, this attack opens the door for much sophisticated form of attacks, such as Man-in-Middle attack or even domain name system spoofing and many more sophisticated forms of attacks. In this paper, we will explore different tools and methods used in detection and prevention of Address Resolution Protocol Spoofing attack. The ARPspoofer tool will be used to send spoofed Address Resolution Protocol reply packet to a host in local area network to further study how a host maintains the address resolution protocol cache table with the spoofed or a fake media access control, MAC cache table. Finally, we will compare the tools and methods used in detection and prevention against Address Resolution Protocol Spoofing attack in terms of their effectiveness in detection and prevention of the attack and system performance requirements.

Keywords: *ARP Spoofing, ARP Spoofing Attack, Packet Sniffing, ARP Inspection, Network Security*

1. INTRODUCTION

The Address Resolution Protocol is the portion of TCP/IP protocol set used for associating Media Access Control destination to the Internet Protocol destination for physical reachability between devices in a network. The Ethernet next hop must be discovered before data encapsulation is accomplished and Ethernet Frame is forwarded to hosts in a local area network.

In order to take place communication in local area network, the lower layer address should be mapped to higher layer address. To make this association, an Address Resolution Protocol is used. Moreover, to avoid unnecessary generation of additional broadcast network traffic and to improve network performance an Address Resolution Protocol is required. Data link forwarding relies on the knowledge of the MAC address of the data link destination. The

source must be aware of the target MAC address to which data should be transmitted.

To transmit data at a data link layer, the forwarding host should have information of the lower layer address of the destination device to which Ethernet frame is to be forwarded. If the ARP cache table of a device is empty, then the device has to learn or resolve the upper layer address to lower layer address to physically reach the destination of the device [1, 2, 3]. The device learns the Media Access Control address through the Address Resolution Protocol process. Address Resolution Protocol process is required to resolve the Media Access Control address, to the internet protocol address.

When a device needs to transmit any network traffic to another device for the very first time, the address resolution process is required. The device broadcasts an Address

Resolution Protocol packet to every other device in the network. Upon receiving the Address Resolution Protocol broadcast traffic, a device whose Internet Protocol address matches with the destination header in the broadcasted Address Resolution Protocol packet, reply's to the sender. This reply packet contains the hardware address of the device. Through this process a device learns the hardware address of all devices in the network [4].

The Address Resolution Protocol is very important for physical reachability between devices in Local Area Networks. This protocol does not support validation, and this is often used as part of other serious attacks such as Resolution Protocol spoofing, Distributed Denial of Service (DDoS) attack. In Resolution Protocol process spoofing, the spoofer can capture the packets between devices in the network. In Denial of Service violence, the invader marks a target device reject communication with other device. Due to this reason Address Resolution Protocol spoofing remains a bad attack in the local area network [5, 6].

In Address Resolution Protocol spoofing the attacker marks Media Access Control table as its target. In local area networks, a device may connect with another device such as a router, to do so; the device transmits a Media Access Control address. In transmission of the lower layer address, reaching the intended destination relies on firstly resolving the Media Access Control address to the Internet Protocol address of a device. There is no verification or authentication procedure for the Address Resolution Protocol. Any device that needs to reach another device on a local area network creates an Address Resolution Protocol packet with destination address of broadcast Media Access Control which is FF-FFF-FF-FF-FF-FF [7-10].

The Address Resolution Protocol process request/reply between two hosts is shown in figure 1 and is discussed in detail in the upcoming sections. When the host-1 needs to communicate with the host-2 which is in the same network, does the following:

- 1) Sends ARP request to host-2 with upper layer protocol address, MAC address as a source, the upper layer protocol address and a broadcast MAC address as destination address.
- 2) The host compares the upper layer address in the received frame with the upper layer address. Since, the frame was destined to this host, the

frame is processed and ARP reply packet is sent as a unicast frame with the host-1 MAC address as destination and host-2 MAC address as source in the frame.

3) Upon receiving this frame, host-1 will associate the new MAC received in frame in step 2 to upper layer protocol address and updates the ARP cache as shown in figure 3.

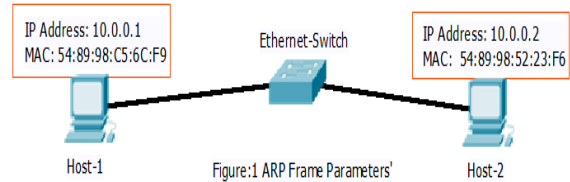
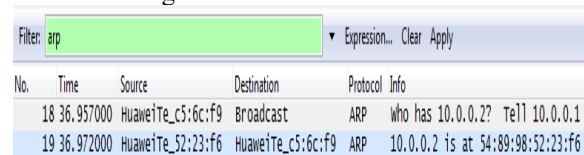


Figure 1: ARP process

When host-1, wants to communicate with host-2, host-1 will send ARP broadcast frame using its MAC address and IP address in the ARP request frame, that is the pair <10.0.0.1, 54-89-C5-6C-F9> as a source in the frame header, and the pair <10.0.0.2, FF-FF-FF-FF-FF-FF> as destination in the frame header to host-2 as shown in figure 2.



| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|-------------------|-------------------|----------|----------------------------------|
| 18 | 36.957000 | HuaweiTe_c5:6c:f9 | Broadcast | ARP | who has 10.0.0.2? Tell 10.0.0.1 |
| 19 | 36.972000 | HuaweiTe_52:23:f6 | HuaweiTe_c5:6c:f9 | ARP | 10.0.0.2 is at 54:89:98:52:23:f6 |

Figure 2: host-1 ARP request process

Upon receiving the frame, Host-2 will compare the layer-3 address, that is the IP address in the received frame with its' own layer-3 address that is, 10.0.0.2 is compared with 10.0.0.2 and as the address is the same as Host-2's IP address, the host will reply to Host-1. The reply packet will contain the following network parameters in the frame header. The source address in the frame header will be the pair <10.0.0.2, 54-89-98-52-23-F6> and the destination address is the pair <10.0.0.1, 54-89-C5-6C-F9>. Now since Host-1 received ARP reply packet with the source MAC address, the host updates its' ARP cache as shown in the diagram below.

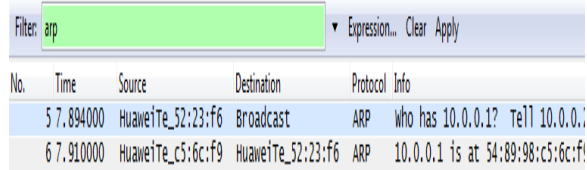
```
PC>arp -a
```

| Internet Address | Physical Address | Type |
|------------------|-------------------|---------|
| 10.0.0.2 | 54-89-98-52-23-F6 | dynamic |

Figure 3: host-1 ARP cache

When Host-2, wants to communicate with Host-1, it will send ARP broadcast frame using

its own MAC address and IP address in the ARP request frame, that is the pair <10.0.0.2, 54-89-98-52-23-F6> as a source in the frame header, and the pair <10.0.0.1, FF-FF-FF-FF-FF-FF> as destination in the frame header will be sent to Host-1 it is shown in the figure 4.



| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|-------------------|-------------------|----------|----------------------------------|
| 5 | 7.894000 | HuaweiTe_52:23:f6 | Broadcast | ARP | Who has 10.0.0.1? Te11 10.0.0.1 |
| 6 | 7.910000 | HuaweiTe_c5:6c:f9 | HuaweiTe_52:23:f6 | ARP | 10.0.0.1 is at 54:89:98:c5:6c:f9 |

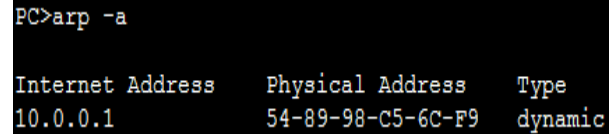
Figure 4: host-2 ARP request process

Upon receiving the frame, Host-1 will compare the layer-3 address, that is the IP address in the received frame with its' own layer-3 address that is, 10.0.0.1 is compared with 10.0.0.1 and as the address is the same as host-1's IP address, the host will reply to host-2. The reply packet will contain the following network parameters in the frame header. The source address of the frame header will be the pair <10.0.0.1, 54-89-C5-6C-F9> and the destination address is the pair <10.0.0.1, 54-89-98-52-23-F6>. Now since host-2 has received ARP reply frame from host-1 with <IP, MAC address> pair that is <10.0.0.1, 54-89-98-52-23-F6> in the frame header, host-2 updates the ARP cache as shown in figure 5.

2. RELATED WORKS

In this section, we will focus on the research papers [16-21] related to address resolution protocol spoofing attack detection and prevention.

DDoS attack analysis and study [16] provides different methods used in prevention and detection of address resolution protocol spoofing, are: use gateway DDoS violence stoppage; hardening the Trusted Platform Module; system defense enhancement. The gateway DDoS violence stoppage can avoid man-in-the-middle attack by using ARPSpoof as a means of making the attack. If we stop gateway violence, then no attacker can capture the traffic destined to this gateway. Hardening the system can also prevent ARPSpoofing attack; the system can be hardened by using static Address Resolution Cache table instead of dynamic Address Resolution Cache table.



```
PC>arp -a
```

| Internet Address | Physical Address | Type |
|------------------|-------------------|---------|
| 10.0.0.1 | 54-89-98-C5-6C-F9 | dynamic |

Figure 5: host-2 ARP cache

The ARP request/reply is performed as discussed in the introduction section and subsequent communication the hosts, h1 and h2 shown in figure 1, use their ARP cache to make forwarding decision upon receiving any frame in the local area network, LAN.

The Address Resolution Protocol forward by using FF-FF-FF-FF-FF-FF as the destination hardware address to reach each device on the network, requesting that a specific Internet Protocol destination respond with the matching Media Access Control address. When a device responds with a Media Access Control destination data can be delivered to that Media Access Control destination, the problem is that the replay is acknowledged without authentication. Address Resolution Protocol spoofing involves transmitting fake packets to the target device so they wrongly associate the spoofer's Media Access Control destination that belongs to the Internet Protocol destination [11-15].

The network traffic sniffers [17] are used to sniff un-encrypted confidential information in transit without the knowledge of the owner. For example e-mail credentials or the parameters required in Telnet sessions can be sniffed using wireshark. Wireshark, tcpdump and ettercap, can be used to sniff packets. Some sniffers only sniff packets with no inspection; hence the researchers must sniff packets with sniffers which support better inspection of packets to better understand the packet characteristics in their work.

In this work the packets selection is made possible by the user with the help of the tools that inspect packets and sniff packets and these selected packets will be stored in the main memory. This may reduce the wastage of memory by storing every packet in to the memory.

Detection of Address Resolution Protocol spoofing using Internet Control Message standard [18] suggested a method to probe the legitimacy of every active Address Resolution Protocol packet without sniffing every packet in the network. This is achieved by sending an Internet Control Message Protocol ping packet on behalf of every fooled Address Resolution Protocol packet on the network to infer with validity of the reply. The method detects true Internet Protocol-Media Access Control destination associations of genuine device and the attackers' device during a genuine spoofing.

The Internet control message protocol probing for authentication purpose creates unnecessary traffic overload on the network. Packet analyzers are used in packet inspection, packet capturing, ARP spoofing attack detection, prevention and ethical network monitoring. Examples include ettercap and colasoftware [19].

The comparison between network traffic analyzer programs like wireshark and tcpdump reveals that wireshark uses lots of power, main storage space and CPU time. As compared to tcpdump, wireshark makes better capability for inspection of sniffed packet [20].

For continuous network traffic monitoring systems wireshark is preferred in discovery of spoofed address resolution protocol network traffic and notifies instantly.

As the researcher having the thorough understanding of network traffic and traffic sniffing the tcpdump represents data in preferred format. In other cases, wireshark would be the best choice. Where the requirement is network traffic capturing using hosts with low storage space and energy, tcpdump is preferred for monitoring and capturing network traffics. In the case the devices with high storage capacity, processing power and energy wireshark is better choice for network traffic capturing and inspection of the network traffic. Some of the tools used in detection and prevention of Address Resolution Protocol Spoofing attack are; ARPwatch, XARP, Static ARP Table, ARPAlert, wireshark and ARPon [21].

3. METHODS USED IN DETECTION AND PREVENTION AGAINST ARP SPOOFING ATTACK

In this section we will focus on the methods used in detection and prevention of Address Resolution Protocol spoofing attacks, such as XARP, Wireshark, ARPAlert, ARPon and static inspection. One of the best methods to deal with Address Resolution Protocol spoofing attack is static inspection. In this method, media access control address, MAC is statically mapped to specific Internet Protocol address and the mapping is controlled by administrator manually. Using this method, highest level of Local Area Network security can be achieved but, it is not feasible in larger network with hundreds of hosts. Hence, automated tools are required to prevent the Address Resolution Protocol Spoofing attack in Local Area Networks and these tools are discussed in detail in upcoming sections.

3.1. XARP

XARP is an application used for Address Resolution Protocol Spoofing attack detection. This runs on windows, Mac or UNIX systems. It requires very limited memory space and processing time. It is valuable easily usable software to screen the Address Resolution Protocol table of a computer. The XARP sends instant packets to a computer Address Resolution Protocol table on the computer on which it is running. If any change occurs with the previously stored address resolution table of the computer and received address resolution protocol reply packet, it notifies the change in <Internet Protocol address, Media Access Control address> matching Address Resolution Protocol reply packet to the previous one. Thus, XARP is required for detecting Address Resolution Protocol spoofing attack. The fake address resolution protocol replay detected by XARP is shown in Figure 7.

XARP raises alert in cases where Address Resolution Protocol spoofing attack is attempted. In figure 1, we have sent Address Resolution Protocol frame replay to a host with Internet Protocol address 10.0.0.1 as a target or victim.

```
tt@ubuntu:~$ sudo arpspoof -i ens33 -t 10.0.0.1 -r 10.0.0.2
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.1 is-at 0:c:29:79:e1:8b
0:c:29:79:e1:8b 0:50:56:c0:0:8 0806 42: arp reply 10.0.0.2 is-at 0:c:29:79:e1:8b
```

Figure 6: Address Resolution Protocol replay packets

As illustrated in figure 6, a command line arpspoof tool was used to send address resolution protocol reply packet to a host destined at upper layer protocol address which is the Internet Protocol address of 10.0.0.2.

Upon receiving this packet, the host maintains its media access control address with the received MAC address in the arpspoof attack option.

| IP | MAC | Host | Vendor | Interface | Online | Cache | First seen | Last seen |
|----------------|-------------------|-----------------|--------------|---------------------------------------|---------|-------|--------------------|--------------------|
| 10.0.0.1 | 00-0c-29-79-e1-8b | DESKTOP-1031E7U | Vmware, Inc. | One - VMware Virtual Ethernet Adapter | unknown | no | 9/27/2010 07:23:30 | 9/27/2010 07:52:07 |
| 10.0.0.2 | 00-50-56-c0-0-08 | 10.0.0.2 | Vmware, Inc. | One - VMware Virtual Ethernet Adapter | unknown | no | 9/27/2010 07:23:37 | 9/27/2010 07:52:07 |
| 10.128.0.0 | 00-0c-29-79-e1-8b | 10.128.0.0 | Vmware, Inc. | One - VMware Virtual Ethernet Adapter | unknown | yes | 9/27/2010 07:23:38 | 9/27/2010 07:43:23 |
| 10.255.255.254 | 00-50-56-c0-0-08 | 10.255.255.254 | Vmware, Inc. | One - VMware Virtual Ethernet Adapter | unknown | yes | 9/27/2010 07:23:38 | 9/27/2010 07:38:50 |

Figure 7: Address Resolution Protocol spoofing attack detection by XARP

Figure 2 shows Address Resolution reply packet that is received from the attacker’s machine. When the reply packet was received from the attacker as shown in figure 1, the XARP tool raises alert indicating the received reply packet. XARP simply detects address resolution protocol spoofing attack and it does not have any mechanism to block the attacker’s from being modifying the victims Media Access Control address.

3.2. DYNAMIC INSPECTION

The dynamic Address Resolution Protocol Inspection detects and stops Address Resolution

Protocol Spoofing, Address Resolution Protocol Cache Poisoning (ARPCP) attacks, and associated attacks against these attacks and the resulting attacks, such as capturing the network traffic or packets, session hijacking, and SQL injections. Dynamic Address Resolution Protocol attack is enabled on the network interface ens33 and shown in Figure 8, it accepts or ignores address resolution protocol replay packets received from any host in the network, depending on the configuration file.

```
tt@ubuntu:~/Desktop$ sudo arpon arpalert.darpi -i ens33 -y 1 -D
[sudo] password for tt:
07:26:01 WAIT LINK on ens33...
07:26:01 DARPI on
DATE = <09/27/2018>
DEV = <ens33>
HW = <0:c:29:79:e1:8b>
IP = <10.128.0.0>
07:26:04 ARP cache, ACCEPT
src HW = <0:50:56:f8:84:5c>
src IP = <10.0.0.2>
07:26:59 ARP cache, ACCEPT
src HW = <0:50:56:c0:0:8>
src IP = <10.0.0.1>
07:27:11 ARP cache, ACCEPT
```

Figure 8: Dynamic Address Resolution Protocol Inspection

3.3. STATIC INSPECTION

The static Address Resolution Protocol Inspection detects and blocks Address Resolution Protocol Spoofing attack. In this method the system is configured with allowed and disallowed list of media access control, internet Protocol address pairs. When address resolution protocol reply packet is received from any host in the network, it will be compared with the configuration file, if the replay matches with the configuration file parameters then, address resolution protocol table is updated accordingly and blocked otherwise. In figure 11 the static Address resolution protocol inspection method is shown.

```
tt@ubuntu:~/Desktop$ sudo arpon arpalert.sarpi -i ens33 -s
07:33:25 WAIT LINK on ens33...
07:33:25 SARPI on
    DATE = <09/27/2018>
    DEV = <ens33>
    HW = <0:c:29:79:e1:8b>
    IP = <10.128.0.0>
    CACHE = </etc/arpon.sarpi>
07:33:29 ARP cache, IGNORE
    src HW = <0:50:56:f8:84:5c>
    src IP = <10.0.0.2>
07:34:00 ARP cache, IGNORE
    src HW = <0:50:56:c0:0:8>
    src IP = <10.0.0.1>

tt@ubuntu:~/Desktop$ sudo arp -s 10.0.0.4 00:50:56:c0:0:8
[sudo] password for tt:
tt@ubuntu:~/Desktop$ sudo arp -s 10.0.0.2 00:50:56:c0:0:8
tt@ubuntu:~/Desktop$ arp
Address          HWtype HWaddress      Flags Mask      Iface
10.0.0.2         ether   00:50:56:c0:0:08 CM              ens33
10.255.255.254   (incomplete)
10.0.0.4         (incomplete)   ens33
```

Figure 9: Static Address Resolution Protocol Inspection

3.4. STATIC ADDRESS RESOLUTION TABLE

The static Address Resolution Protocol table can be maintained by the network administrator but in today's very large and complex networking environment, it is almost impossible for administrator to assign a static Media Access Control-Internet Protocol entry for every device that is connected to the network. Secure distribution of tables not possible. Depending on operating system version static Address Resolution Protocol entries are being overwritten by the dynamic one.

Arp command with option -s [Internet Protocol address, Media Access Control Access] is used to set up a new Address Resolution Protocol table entry statically.

```
tt@ubuntu:~$ arp
Address          HWtype HWaddress      Flags Mask      Iface
10.0.0.5         ether   00:50:56:c0:0:01 CM              ens33
```

Figure 10: Address Resolution Protocol cache table

Using the command `arp -s 10.0.0.2 00:50:56:f8:84:5c` we can add `<10.0.0.2, 00:50:56:f8:84:5c>` into the Address Resolution Protocol table of a host in network. As a result of issuing the arp command, a new static entry is maintained by the host as shown in figure 10.

```
tt@ubuntu:~$ sudo arp -s 10.0.0.2 00:50:56:f8:84:5c
tt@ubuntu:~$ ip neighbour show
10.0.0.1 dev ens33 lladdr 00:50:56:c0:00:08 STALE
10.0.0.2 dev ens33 lladdr 00:50:56:f8:84:5c PERMANENT
10.255.255.254 dev ens33 lladdr 00:50:56:ee:ff:b9 STALE
```

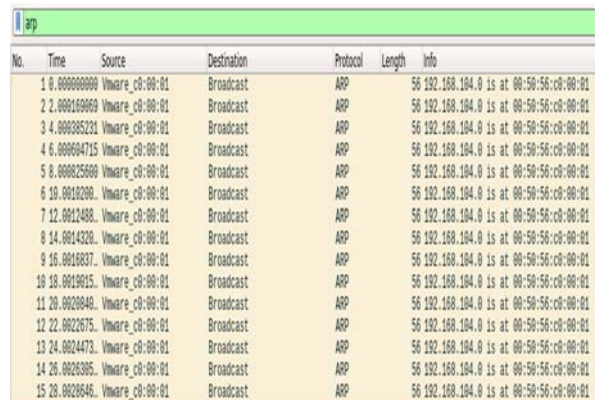
Figure 11: Static Address Resolution Protocol

The entry `<10.0.0.5, 0:50:56:C0:00:01>` as shown in figure 6 is static entry in the Address Resolution Protocol table. The Media Access Control address, `00:50:56:f8:84:5c` is permanently mapped to the upper layer address which is the Internet Protocol address of a host `10.0.0.2`. Due to this static permanent mapping, any reply packet received by this host in attempt to modify this mapping will no longer accepted by the device.

3.5. WIRESHARK

Wireshark can be used to interactively dump and analyze network traffic what's going on inside the wireless medium, examine the address resolution protocol replay packets, their source and destinations. It can be used for the following purposes;

- To examine the packets passing through the entire Wireless Local Area Network.
- Detect Address Resolution protocol spoofing attack.
- Capture live packet data from a network interface.



| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------------|-------------|----------|--------|---------------------------------------|
| 1 | 0.00000000 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 2 | 0.00169969 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 3 | 0.00339938 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 4 | 0.00509907 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 5 | 0.00679876 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 6 | 0.00849845 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 7 | 0.01019814 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 8 | 0.01189783 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 9 | 0.01359752 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 10 | 0.01529721 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 11 | 0.01699690 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 12 | 0.01869659 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 13 | 0.02039628 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 14 | 0.02209597 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |
| 15 | 0.02379566 | Vmware_c0-00:01 | Broadcast | ARP | 56 | 192.168.104.0 Is at 00:50:56:c0:00:01 |

Figure 12: Wireshark ARP packet analysis

As shown in Figure 12, the host machine which is the attacker's machine sends Address Resolution Protocol packets continually. But normal flow of Address Resolution Protocol request reply is different from the one given above. No host has requested Address Resolution Protocol, but reply packet is sent even if the host does not send any Address Resolution Protocol packet. And therefore, by sending this forge Address Resolution Protocol reply packet to a particular host in local area network, Local Area Network, we can update the Address Resolution Protocol cache of the host. This is the first step in Address Resolution Protocol spoofing attack.

3.4. COMPARISON OF ARP SPOOFING DETECTION AND PREVENTION TOOLS

To evaluate Address Resolution Protocol spoofing attack detection and prevention methods as a proof-of-concept, we have used ARPspooftool that simulated Address Resolution Protocol spoofing behavior and attempted to detect it using XARP and ARPawtch, wireshark, arpalert and ArpON tools which we have described in the previous sections. In the following sections we will discuss memory and CPU usage of these methods.

```
tt@ubuntu:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
arpalert  1522  0.0  0.4 21536 4004 ?        S    21:19   0:00 /usr/sbin/arpa
```

Figure 13: system resource usage by ARPALERT ARP spoof detection tool

ARPAAlert consumes less CPU time and 0.2 % of the system memory as shown in figure 13. It is not platform independent run on UNIX platform only.

```
tt@ubuntu:~$ pidof wireshark
3251
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1879 root  20  0 485904 36924 9064 S 18.7 3.7 0:31.52 Xorg
2170 tt    20  0 1236220 81292 20200 S 15.4 8.2 0:46.81 compiz
3587 tt    20  0 633992 29556 24636 S 6.9 3.0 0:00.58 gnome-scre+
2884 root  20  0  0 0 0 S 1.0 0.0 0:02.01 kworker/0:2
3251 root  20  0 761600 127384 14732 S 1.0 12.8 0:06.48 wireshark
```

Figure 14: System memory and CPU usage by Wireshark packet inspection tool

Wireshark is resource intensive, platform independent tool used in detection of Address Resolution protocol spoofing attack.

```
tt@ubuntu:~$ pidof arpwatc
2780
tt@ubuntu:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      2780  0.0  0.4 11344 4464 ?        S    12:30   0:00 arpwatc
```

Figure 15: ARPWATCH Memory usages, ARP spoofing attack detection and prevention tool

As shown in the figure 15, ARPWatch uses 0.1% of the CPU time and it is a light weight program. The memory space requirement is 0.4MB in % as shown in figure 15 and it can be used on a system with less processing power and storage space. This test is carried on UNIX system and it is not compatible to Windows platform.

| Name | CPU | Memory | Disk | Network |
|---------------|------|--------|--------|---------|
| Xarp (32 bit) | 0.3% | 4.2 MB | 0 MB/s | 0 Mbps |

Figure 16: System Memory and CPU usage by XARP, ARP spoof detection tool

As shown in the figure 16, XARP uses 0.3% of the CPU time and it is a light weight program. The memory space requirement is 4.2MB in % as shown in figure 16 and it can be used on a system with less processing power and storage space.

3.5. COMPUTATIONAL RESOURCE USAGE OF ARP SPOOF DETECTION AND PREVENTION METHODS

The memory space requirement of ARP watch, Wireshark, XARP and Arpalert ARP is shown in Figure 17.

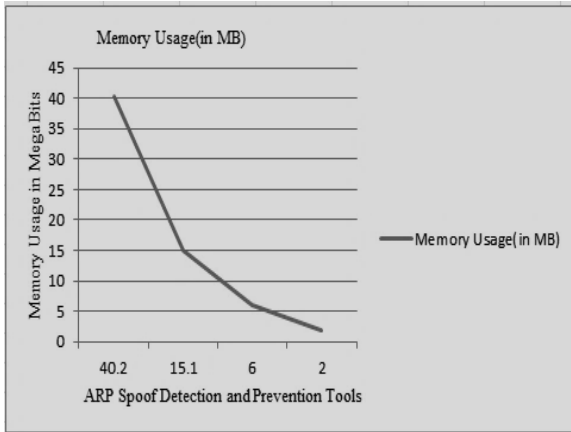


Figure 17: Memory usage by ARP attack detection tools

In figure 17, the system memory usage by different Address Resolution Protocol attack detection and prevention tools is shown. As illustrated in figure, Wireshark is a resource intensive, compared to all other methods and tools used to deal with Address resolution protocol spoofing and prevention. TCPDUMP is the better in terms of memory usage and XARP is moderate in memory usage.

CPU time consumption by ARP watch, Wireshark, XARP and Arpalert is shown in Figure 18.

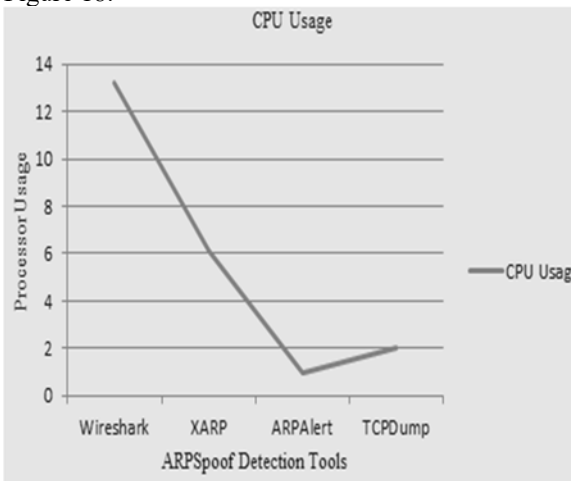


Figure 18: processing efficiency of ARP attack detection tools

Wireshark requires high performing systems, as it consumes higher processing time compared to ARPalert, and XARP. XARP is moderate in performance and may run effectively on systems with less processing power. ARP alert is the best in terms of processor usage and hence in performance.

3.6. COMAPATIBILITY AND PLATFORM INDEPENDENCE

In this section we will compare address resolution protocol detection methods in terms of their cross platform independence as shown in Table 1.

Table 1 Computing platforms support of ARP spoofing detection methods/tools

| ARP spoof Detection methods | Computing platform | | |
|-----------------------------|--------------------|-----|------|
| | Windows | Mac | Unix |
| 1. XARP | √ | √ | √ |
| 2. Arpalert | X | X | √ |
| 3. ArpON | X | X | √ |
| 4. Wireshark | √ | √ | √ |
| 5. TCPdump | X | X | √ |

It is shown in table 1 that wireshark and XARP are platform independent but other tools are not platform independent as they are suited to only UNIX system.

3.7. TCPDUMP

Tcpdump prints out descriptions about the contents of packets on a network interface. It is a packet capturing tool which can be used in detection of Address Resolution Protocol spoofing attack on a system connected to the local area network. Address Resolution Protocol spoofing attack detection process by tcpdump is shown in Figure 8.

```
root@ubuntu:/home/tt# tcpdump -i ens33
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
12:48:56.571689 ARP, Request who-has 10.0.0.2 tell 10.128.0.0, length 28
12:48:56.572277 ARP, Reply 10.0.0.2 is-at 00:50:56:f8:04:5c (oui Unknown), lenat
```

Figure 19: tcpdump Address Resolution Protocol packet analysis

Figure 19 shows that ARP spoof attack detected by TCPdump on a network interface ens333. A simple command line tool tcpdump was used along with an option, to specify the interface and interface name ens33. The tool monitors or inspects every packet received by this interface and displays any unknown or un-requested ARP reply packets.

3.8. DISCUSSIONS

Address Resolution Protocol spoofing attack which is the most underestimated attack, can be a base for more sophisticated attacks.

Although poisoning the Address Resolution Protocol table of a host will do nothing when it is in the inactive state, this kind of attack may lead to loss of sensitive data, disclosure of confidential information to the intruder as the Address Resolution Protocol spoofing is the basis for other advanced attack.

To mitigate this attack different software tools were developed. Among these we have discussed the most popular open source Address Resolution Protocol spoofing attack detection and prevention tools such as XARP, ARPwatch, Wireshark and the static method. Those all methods may be useful based on the level of security requirement and the nature of a company network.

For example in a network consisting of three or four computer or in small workgroup the static method may be preferred than all other methods.

The main solutions that are proposed to detect Address Resolution Protocol spoof attacks are XARP, static method and ARPwatch. In this section we have discussed different Address Resolution Protocol spoof detection and prevention methods such as XARP software tool for detection of Address Resolution Protocol spoofing, static <Internet Protocol, Media Access Control> entry and ARPwatch.

Table 2 ARP Spoofing attack detection and prevention tools

| ARP spoof detection and prevention tools | System Resource requirement | Detection and inspection |
|--|-----------------------------|---|
| Wireshark | High | Powerful in detection but, requires manual inspection |
| ARPwatch | Medium | Powerful in detection but, requires manual inspection |
| XARP | Moderate | Powerful in detection but, requires manual inspection |
| ARPalert | Low | Powerful in detection and does not, require manual intervention |

4. OPEN RESEARCH ISSUES

An Address Resolution Protocol implementation considers only normal network service without verifying improper service interaction or malicious behaviors in the network. For example, after receiving Address Resolution Protocol response packets, hosts do not verify whether they have sent the Address Resolution Protocol request, but directly replace the original Address Resolution Protocol cache table with the mapping between MAC and IP addresses in the response packet.

An Address Resolution Protocol does not have authentication mechanism by which a host or hosts sending reply packet can authenticate themselves to the host receiving the reply packet. Adding authentication information to the address resolution protocol packet and verifying improper or malicious address resolution protocol replay packet in a network is already a research issue.

5. CONCLUSION

In this paper, we have analyzed Address Resolution Protocol spoofing attack and packet sniffing tools. Address Resolution Protocol Spoof was used to generate spoofed Internet Protocol-Media Access Control association and it is transmitted to a host in wireless local area network, forcing the receiving host to update the Address Resolution Protocol table with a spoofed Internet Protocol-Media Access Control association. We have compared different methods and tools used in detection and prevention against Address Resolution Protocol Spoofing attacks in terms their effectiveness in detection and performance.

Wireshark requires larger memory space, higher system performance but, it is very powerful in detection of fake Address Resolution replay packets and does not have a mechanism to block the reply packet form being received by the victim machine, as a result, manual inspection is required by the network administrator. Another perfectly performing tool is ARPalert which is applicable to small high security networks, such as Wireless Local Area Networks with two or more dozen machines. Compared to Wireshark and ARPalert, XARP has a better performance, it is a lightweight program yet very powerful in detection of Address Resolution Protocol spoofing attack. Unlike, the ARPalert which runs only on UNIX,

XAPP is platform independent and runs on UNIX and Windows systems.

The other most important tool used in Address Resolution Protocol spoofing attack is the ARPWatch tool. This tool runs on UNIX system. It can be configured on UNIX system to automatically detect any spoofed Address Resolution Protocol reply packet. This tool is more powerful in detection and can be automated so that once configured it does not need human intervention to deal with Address Resolution Protocol spoofing attack.

REFERENCES

- [1] Zhang Chao-yang, DDOS attack analysis and study of new measures to prevent ARP Spoofing attack (Communications Magazine, *IEEE*, 40(10), pp.42-51, Oct (2012).
- [2] Asrodia, Pallavi, and Hemlata Patel. "Analysis of various packet sniffing tools for network monitoring and analysis." *International Journal of Electrical, Electronics and Computer Engineering* 1.1 (2012): 55-58.
- [3] Nedhal A. Ben-Eid(2015), Gao Jinhua and Xia Kejian, Detection of ARP Spoofing Attack Using ICMP Protocol, *IEEE*, (2013).
- [4] Sherin Hijazi, Mohammad S. Obaidat, A new detection and prevention system for ARP spoofing attack using static entry, *IEEE*,2018.
- [5] Jami Manzoor, Avinash Kumar, Sweety Kumari, ARP SPOOFING AND MAN IN THE MIDDLE ATTACK, , *International Journal of Computer Engineering and Applications*, Volume X, Special Issue, ICRTCST -2016.
- [6] Prerna Arote, Karam Veer Arya, Detection and Prevention against ARP poisoning attack using modified ICMP and voting, international conference on computational networks, *IEEE*, 2015.
- [7] J. David Brown, Tricia J. Willink, A new look at: ARP spoofing to create routing loops in Ad Hoc networks, *IEEE*, 2018.
- [8] D Raviya Rupal, Dhaval Satasiya, Hires Kumar, Archit Agrawal, Detection and prevention of ARP poisoning in dynamic IP configuration, *IEEE*, 2016.
- [9] S.Venkatramulu,, Dr.C.V Guru Rao Various Solutions for Address Resolution Protocol SPOOFING ATTACK, *International Journal of Scientific and Research, publications*, Volume 3, Issue 7, July 2013-1 -ISSN 2250-3153.
- [10] Ravi Raj Saini ,Himanshu Gupta,A security framework against Arp spoofing, international conference on reliability, infocom technologies and optimization trends and future directions, *IEEE*, 2015.
- [11] Ai-zeng Qian, "The Automatic Prevention and Control Research of ARP Deception and Implementation," 2009 WRI World Congress on Computer Science and Information Engineering, pp. 555-558, April 2009.
- [12] Boughrara, A.; Mammari, S., "Implementation of a SNORT's output Plug-In in reaction to ARP Spoofing's attack," 2012 6th International Conference on Sciences of Electronics Technologies of Information and Telecommunications (SETIT), pp.643,647, 21-24 March 2012.
- [13] Ferdous A. Barbhuiya ; Santosh Biswas ; Neminath Hubballi ; Sukumar Nandi, A Host based DES approach for detecting ARP spoofing attack, 2011, IEEE Symposium n computational intelligence on cybernetics.
- [14] Xiangning HOU, Zhiping JIANG, and Xinli TIAN, "The detection and prevention for ARP Spoofing based on Snort," 2010, IEEE.
- [15] Ahmed M.AbdelSalam, Wail S.Elkilani, Khalid M.AminAn Automated approach for Preventing ARP Spoofing Attack using Static ARP Entries, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, 2014.
- [16] Andre P. Ortega, Xavier E. Marcos, Luis D. Chiang and Cristina L. Abad, " Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt," Latin American Network Operations and Management Symposium. (LANOMS), pp. 1- 9, Oct. 2009.
- [17] Prerna Arote, Karam Veer Arya , Detection and Prevention against ARP poisoning attack using modified ICMP and voting, 2015, IEEE, 2015 International Conference on Computational Intelligence and Networks.

- [18] Nadhal A. Beneid, Ethical Network Monitoring Using Wireshark and Colasoft Capsa as Sniffing Tools,2015.
- [19] Gao Jinhua, Xia Kejian, detection of ARP spoofing attack using ICMP protocol, IEEE, 2013.
- [20] Asrodia, Pallavi, and Hemlata Patel, network traffic analysis using packet sniffer, International Journal of Engineering Research and Applications, 2012.
- [21] M. K. Chirumamilla and B. Ramamurthy, “Agent based intrusion detection and response system for wireless LANS,” in ICC '03. IEEE International Conference on Communications, 2003, pp. 492–496.