ARP Poisoning Detection and Prevention using Scapy

Aayush Majumdar, Shruti Raj and T.Subbulakshmi
School of Computer Science and Engineering, Vellore Institute of Technology
Chennai, Tamil Nadu, India
E-mail: aayush.majumdar2018@vitstudent.ac.in

Abstract. Address Resolution Protocol is a protocol associated with mapping a given IP address with the associated MAC address. ARP Poisoning or Spoofing attack is an attack which carried out over a Local Area Network. In an ARP Poisoning/Spoofing attack malicious ARP Packets are sent to a default gateway on LAN with intent to change the IP address - MAC address pairings in the ARP cache table. ARP Spoof attack tool has been developed with a script written in Python Programming language using scapy library. A detection algorithm has been proposed for detecting the above generated ARP Poisoning attack (or any ARP Poisoning attack in general) and implemented the same using a python script with scapy library. The detection algorithm is based on analyzing the real MAC Address and response MAC Address of the ARP Packet sniffed for any discrepancies. Lastly, a prevention mechanism is proposed and implemented for ARP Poisoning attacks by implementing static entries in ARP table.

1. Introduction
Address Resolution Protocol (ARP) is a basic and amongst some of most commonly used protocols pertaining to computer communications. In a Local Area Network, ARP messages work by resolving IP addresses into corresponding MAC or physical addresses. Address Resolution Protocol associated with mapping a given IP address with the associated MAC address. However, ARP protocol has some limitations which make it vulnerable and makes it easier for attackers to exploit. The most eminent limitations are – stateless and unauthenticated nature of ARP. These limitations can be easily exploited by attackers for their personal profit.

ARP Poisoning or Spoofing attack is an attack which is carried out over a Local Area Network. In an ARP Poisoning/Spoofing attack malicious ARP Packets are sent to a default gateway on LAN with intent to change the IP address - MAC address pairings in the ARP cache table.

The ARP Poisoning attack consists of a false ARP reply message sent by the attacker to the default gateway which informs that the MAC address of the attacker should be associated with the victim’s (target’s) IP address and vice versa. Once the message has been received by the default gateway, it broadcasts its changes to all the other devices connected to the network. Now, all of the target’s (victim’s) traffic which is sent to any other device on the network travels through the attacker’s computer. Thus, the attacker can inspect or modify the messages before forwarding it to real destination. Therefore, ARP Spoofing (ARP Poisoning) can be done to carry out Man In The Middle attacks.

Other than Man-in-the-Middle Attacks, denial-of-service condition over a LAN can also be caused by ARP Spoofing by simply intercepting or dropping and not forwarding the target's packets.

The tools discussed were developed with python scripts using Scapy to simulate ARP Poisoning attacks by exploiting the loopholes and performing a Man-In-The-Middle (MITM) attack. The second
part encompasses discussion of developed script which detects ARP poisoning attack and gives a warning. And lastly the proposed solution to prevent the ARP Poisoning attack.

The simulated attack and detection have been performed on VMWare Workstation on Kali Linux environment.

2. Related work

Some pre-existing solutions in this field proposed by other researchers for detection of ARP poisoning attacks and consequently preventing them, focusing on a certain scenario to develop the required tools are shown below.

N Tripathi and Mehtri B M have proposed ICMP based secondary cache approach for the detection and prevention of ARP poisoning [2]. This paper looks into a new efficient approach to avert ARP Spoofing attack. The identical entries of the same MAC address and IP address in the ARP cache table are eliminated from the cache table thus disconnecting the systems with duplicate entries. This uses Secondary Cache which contains the entries with respect to Internet Control Message Protocol (ICMP) responses. IP exhaustion problem is also solved by this method. A single point failure is also mitigated as at every host a secondary cache is maintained which makes this solution a distributed technique.

D Bruschi et al introduce a secure address resolution protocol [3] which enables the user to avoid ARP based attacks. Defense against ARP poisoning is provided by a secure version of ARP where a public/private key pair is provided to all the hosts which is certified by a local trusted party on LAN. Sender digitally signs all the messages to mitigate the injection of poisoned/spooed information.

J Xia et al propose an Active Defense Solution for ARP Spoofing in OpenFlow Network [4]. Software-defined network (SDN) is an emerging new technology which has been rapidly developing for recent years because of its multiple advantages in network management. This paper first implements ARP spoofing in SDN technology and then proposes a defense mechanism for ARP Spoofing in OpenFlow Platform.

D Data introduces the defense against ARP Spoofing attack using Semi-Static ARP cache table [5]. This paper proposes a solution to ARP Spoof attacks using static ARP tables. However, since the process of manually making an entry in ARP Table is tedious this research proposes the automatic ARP cache table management by addition of the ARP validation function. Standard ARP protocol is compatible with this simple technique proposed in the paper.

G Jinhua and Keijian have published work on ARP Spoofing detection algorithm using ICMP Protocol [6]. This paper explores an innovative algorithm using ICMP protocol for detecting poisoned packets sent by attacking hosts that are performing the Spoofing attack. ARP Packets are collected and analyzed and then injected with ICMP echo request packets which probe to find malicious hosts according to its response packets. Activities of other hosts on the network are not disturbed.

Pandey introduces prevention of ARP spofoofing by a probe packet based technique [7]. In this paper, a probe based technique with an Enhanced Spoof Detection Engine (E-SDE) has been proposed which is used to recognize and detect ARP Poisoning attack along with identifying the original MAC, IP address association. ICMP and ARP packets have been used as probe packets. Incremental development of E-SDE has been clearly shown using an attacking model.

Puangpronpitag proposes an innovative, efficient and feasible solution to ARP Spoof problem [8]. In this paper a new solution has been proposed, which is practical, achievable, manageable, effective, cost effective, independent of various platforms and workable with medium and small network sizes alike. Dynamic ARP-spoof protection & surveillance (DAPS) system which is a prototype system is also extensively tested and executed.

Kumar et al introduce a centralized detection and prevention technique against ARP poisoning [9] to manage ARP table entries in all hosts, it proposes ‘Centralized System’ and ‘ARP Central Server ‘(ACS) is used to validate all ARP entries.

S Nam et al propose an enhanced ARP for preventing ARP poisoning-based man-in-the-middle attacks [10]. The mechanism for this concept is based on knowing the correct physical Address for an
IP Address by retaining the correct MAC/IP pair when the machine was alive thus rendering MITM attack unfeasible for that particular IPAddress.

A host based DES approach for detecting ARP spoofing has been proposed by Ferdous A. Barbhuiya et al [11] where they have discussed Discrete Event Based (DES) architecture which does not require changing or modifying existing ARP or violating network layer architecture for detecting ARP Spoofing attacks.

T Chomsiri talks about sniffing packets on LAN without ARP spoofing [12]. The technique aims at analyzing different sniffing data techniques on LAN without the ARP Spoof attack. Two basic programs that is Ethereal and Ping are used.

W Lootah et al introduce Tarp which introduces a Ticket based address resolution protocol [13]. Centrally issued secure Physical/IP address are distributed and attested through existing ARP messages to implement security.

G N Nayak and S Samaddar have proposed a range of variations of man-in-the-middle attack, possible consequences and feasible solutions [14]. Multiple entries for the same MAC address or IP address can be removed from the ARP table using a secondary cache to prevent ARP Spoofing.

Ayad Ibrahim and team have introduced ARP enhancement to Stateful Protocol by registering ARP request [15]. In this method ARP cache poisoning is prevented by enhancing ARP Protocol into Stateful protocol by discarding unregistered replies and based on number of fake replies received by attacker a request is sent.

Design of Win Pcap based ARP Spoofing defense system [16] has been proposed by Mingji, Yang et al. ARP packets capture and analysis is done using Win Pcap driver development monitoring software in switched network inspection system.

V Prevelakis and W Adi have designed a lightweight and secure ARP [17]. DHCP server is used to map IP and MAC Addresses. The computational overhead is very less and it does not affect the reliability of the network.

Talal Alharbi et al introduce securing ARP in software defined networks [18]. New mitigation approach which leverages the Centralized Network Control of SDN by mapping layer 2 and layer 3 protocols is introduced for ARP Spoofing with respect to Software Definition Networks.

Ai-zeng Qian introduces the automatic prevention and control research of ARP deception and implementation [19]. With the given attack mechanisms and ARP deception’s fault phenomenon, in depth analysis from the viewpoint is done and control and automatic prevention algorithms along with model of ARP deception are given.

3. Proposed Architecture

3.1. Environment Setup

For setting up the environment for the project the following tools and libraries are needed:

- VMWare Workstation Pro: VMware Workstation Pro is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems. Using VMWare Workstation Pro, users can set up Virtual Machines on a single device. The Virtual Machines can then be simultaneously used along with the actual physical machine.

- Kali Linux: Designed for digital forensics and penetration testing, Kali Linux is a Debian-derived Linux distribution operating System.

- Scapy: Scapy is a packet manipulation tool for computer networks. Packets can be forged, decoded, sent on wire, captured and requests and replies can be matched using scapy.

Since using ARP poisoning attacks can potentially change or mess up the LAN configurations of the system, the experiments were conducted in an controlled environment using Virtual Machines. The experiment needs two Kali Linux Virtual Machines on VMWare Workstation Pro. One to pose as an attacker and other as the victim. Both Kali Linux VMs should be connected to the same local network so that they have the same gateway and packets can be easily transferred between them. Scapy python
library (usually preinstalled in Kali Linux) should be installed on both the VMs so that the Python Scripts for ARP Poisoning attack and detection can be run.

The tools discussed can be subdivided into three main modules: Attack Generation Module, Detection Module and Prevention Module. The tools discussed were demonstrated using two machines for simulation, one to pose as an attacker and other as the target.

Figure 1 shows the proposed architecture for ARP Poisoning Simulation, Detection and Prevention.

![Proposed Architecture](image)

**Figure 1.** Proposed Architecture

### 3.2. Attack Generation Module

A man-in-the-middle situation is simulated to simulate the attack. The attacker launches an ARP Poison attack which targets the network (or specific hosts) by sending poisoned/false ARP packets, which allows the attacker to intervene, intercept and/or modify the network traffic. Anything and everything that passes through the target’s (victim’s) device can be intercepted by the attacker once he is Man-in-the-Middle.

In a regular network, it is found that devices, in general, communicate first to the gateway and then the internet. As the Man-in-the-Middle, the packets transferred from Target to Gateway and vice versa are first intercepted by the attacker.

The ARP spoof attack simulated using scapy contains the following functions:

- **Obtain MAC Address**: This function is used to obtain the MAC address of any device on the network given the IP address.
- **Spoof**: target IP address’s ARP cache is changed, saying that the device now has the host IP address, that is, i ARP cache table of the victim is changed such that the MAC address associated with the host’s IP address is changed to MAC address of the attacker. It then obtains the Physical/MAC address of the target (victim) and crafts and sends a malicious ARP response.
• Restore: When the ARP Poisoning attack is stopped the original MAC addresses assigned to IP address are needed to be restored. This function re-assigns the real MAC addresses and sends a few legitimate packets before stopping.

3.2.1 Algorithm for Attack Generation
The main idea behind attack generation is to ascertain the ip address of the victim, the medium of attack (in this case the gateway) and to then generate spoofed packets. This can be sequentially represented as follows:
1. Get the MAC Address of host (Gateway) and target (Victim Machine) from their IP Address.
2. Send the ARP response Packets to target using host’s MAC Address (Spoofed)
3. Send the ARP response Packets to host using target’s MAC Address (Spoofed)
4. Continue sending spoofed packets until attack is stopped.
5. Once attack is stopped restore original IP Address MAC Address pair.
6. Send a few legitimate ARP Packets.

Thus, the attacker appears as the host to the target and as the target to the host. Therefore, any messages from target to host (or vice versa) will first be directed to the attacker.

3.3. Attack Detection Module
The basic idea behind the script is to passively scan or monitor the network (keep sniffing the incoming ARP packets). After capturing the ARP packet, it is analyzed to obtain the following two components:
• The source MAC address of the sniffed ARP packet (could be spoofed) .
• The real physical address of the sender (can be found by starting an ARP request for the source IP address).

These two components are then compared. If found to differ, then the system is definitely facing an ARP attack.

The detection algorithm contains the following functions:
• Obtain MAC Address: This function is for obtaining the MAC address of the Gateway (or any other device) given the IP address. This function basically makes an ARP request to the server to obtain the real MAC address of the given IP address.
• Process each sniffed ARP Packet: This function is applied to each packet sniffed. It compares the response MAC address (the address sent by the packet itself) and the Real MAC address obtained using the get_mac function for the given IP. If they are different, a warning message is printed and the packet is discarded.

A script written using Scapy library is used to obtain the response MAC address from the sniffed ARP Packet received by the target machine.

3.3.1. Algorithm for Attack Detection
Thus, detection module focuses on sniffing the incoming packets and processing them to find out if attack has taken place. The algorithm for ARP Poisoning attack detection can be represented as follows:
1. Sniff ARP Packets on network.
2. From given IP Address get the real MAC Address of the gateway.
3. Compare this MAC Address with the response MAC Address of the ARP Packet.
4. If they are the same, sniff the next ARP Packet
5. If they are not the same, generate an alert message showing the Real and the Fake MAC Address and discard the packet.

Thus, ARP Spoofing attack is detected and appropriate warnings about the Real MAC Address and Fake MAC Address is displayed for the user.
3.4 Attack Prevention Module

Setting up a Static ARP entry in the ARP cache table sets up a permanent entry in the ARP cache. This is used as a protection layer against ARP spoofing attacks.

Setting up a static entry for an address prevents the devices from listening to ARP responses for that address and thus ARP spoofing is prevented since the ARP cache of the target cannot be altered for the said MAC address.

Static entries for all the frequently used entries are setup in the ARP Cache table. Any attempt to change the entry in the cache table will disconnect IP Address of the attacker from the network.

4. Implementation details and Results

4.1. Attack Module

Steps to create ARP Spoofer:

1. Use arp command to view the IP addresses of all systems connected to the network. Obtain the IP address of the target (victim) from the ARP Cache Table
2. From the same ARP Cache Table, obtain the physical address corresponding to the IP address (of the host) to be spoofed.
3. Set the target IP (victim IP), Spoof IP and the MAC address (of the host) found in the above steps using the ARP() function to create a spoofed packet.
4. Start the ARP poisoning (spoofing) by sending the spoofed (poisoned) ARP packets to the victim.
5. Display the information of the numbers of packets sent
6. Once the spoofing is stopped, the ARP cache table which now has the spoofed address is reset to original by sending a few legitimate ARP packets to the victim and the host.

4.1.1. Output for attack:

The ARP cache on the victim machine before the ARP Spoofing attack:

```
shruti@kali:~$ sudo arp -a
? (192.168.128.129) at 00:0c:29:39:c9:63 [ether] on eth0
? (192.168.128.2) at 00:50:56:e6:ac:4f [ether] on eth0
```

Here, 192.168.128.128 is IP Address of Victim
192.168.128.129 is IP Address of Attacker
192.168.128.2 is IP Address of Gateway

The ARP cache on the victim machine after the ARP Spoofing attack begins:

```
shruti@kali:~$ sudo arp -a
? (192.168.128.129) at 00:0c:29:39:c9:63 [ether] on eth0
? (192.168.128.2) at 00:0c:29:39:c9:63 [ether] on eth0
```

It is observed that the MAC Address of the gateway has been spoofed to the MAC Address of the attacker. Hence any responses to be sent to the gateway would actually be sent to the attacker.

Running the attack script on the attacker machine:

```
[!]Enabling IP Routing...
[!]IP Routing enabled.
[+]Sent to 192.168.128.128 : 192.168.128.2 is-at 00:0c:29:39:c9:63
[+]Sent to 192.168.128.2 : 192.168.128.128 is-at 00:0c:29:39:c9:63
[+]Sent to 192.168.128.128 : 192.168.128.2 is-at 00:0c:29:39:c9:63
[+]Sent to 192.168.128.2 : 192.168.128.128 is-at 00:0c:29:39:c9:63
[+]Sent to 192.168.128.128 : 192.168.128.2 is-at 00:0c:29:39:c9:63
```
In the above output 192.168.128.2 is IP address of Gateway and 192.168.128.128 is IP Address of the victim. 00:0c:29:39:c9:63 id MAC Address of attacker.

Hence, from the above output, it is observed that to the Gateway, the victim appears to have MAC Address of the attacker and to the victim, Gateway appears to have MAC Address of Attacker. Thus any packets sent between gateway and the victim are first intercepted by the attacker.

4.2. Detection Module

The python script for detection is run on the victim machine. It monitors all the ARP Packets received by the machine. Each ARP Packet has a response MAC Address and an IP Address associated with it. From the IP Address the script obtains the Real MAC Address which should be associated with the IP Address. If the response MAC Address does not match with the Real MAC Address then the script sends a warning stating that the system is under attack.

4.2.1 Output for Attack detection

Output for Detection of ARP Spoof attack is shown in table 1:

| Message Alert: | REAL-MAC:         | FAKE-MAC:         |
|----------------|------------------|------------------|
| ![You are under attack](00:0c:29:39:C9:63) is-at 00:0c:29:39:c9:63 | 00:0c:29:78:B6:E1 | ![You are under attack](00:0c:29:39:C9:63) is-at 00:0c:29:39:c9:63 | 00:0c:29:78:B6:E1 |
| ![You are under attack](00:50:56:E6:AC:4F) is-at 00:0c:29:39:c9:63 | 00:0c:29:78:B6:E1 | ![You are under attack](00:50:56:E6:AC:4F) is-at 00:0c:29:39:c9:63 | 00:0c:29:78:B6:E1 |
From the above results, it is deduced that the detection script detects the ARP spoof attack and identifies the fake MAC Address associated with the IP Address. It also finds what the real MAC Address is.

4.3. Prevention Module
Several methods are available to prevent ARP poisoning, having their own advantages and drawbacks. This paper focuses on static ARP entries as the prevention method for ARP Spoofing.

Using static ARP entries is recommended for smaller networks only as it necessitates a huge administrative overhead. A static ARP entry is added for each system on the network into every individual system.

Allocating the machines with their own set of static IP and MAC addresses helps to prevent spoofing attacks as they can now ignore ARP responses.

An entry in the ARP table can be turned static using the following command:
```
sudo arp -s <IP Address> <MAC Address>
```

Thus, a script for the same is created and added to cronf.

The content of Script is as follows:
```
sudo arp -s 192.168.128.2 00:50:56:e6:ac:4f
```

This Script when applied makes the IP and Mac address static and prevents any spoofing. Any attempt at spoofing will cause the system to automatically disable the attacking device and remove it from the network.

4.3.1 Output For Prevention Using Static Entries
Checking ARP Cache before declaring an entry as static entry:
```
shruti@kali:~$ sudo arp -a
? (192.168.128.129) at 00:0c:29:39:c9:63 [ether] on eth0
? (192.168.128.2) at 00:50:56:e6:ac:4f [ether] on eth0
```

Checking ARP Cache after running script to make static entry:
```
shruti@kali:~$ sudo arp -a
? (192.168.128.129) at 00:0c:29:39:c9:63 [ether] on eth0
? (192.168.128.2) at 00:50:56:e6:ac:4f [ether] PERM on eth0
```

Therefore, the mac address has become “PERM” or permanent and now the target does not listen to ARP responses from Gateway’s IP Address (since it has been declared static) thus preventing ARP Poisoning attack.

5. Conclusion and Scope for Future Enhancement
ARP Poisoning attack allows an attacker to intercept packets on a network, by spoofing its IP address and pretending to be the receiving device, thereby making the victim vulnerable to many other attacks like session hijacking or Denial Of Service. Therefore it is necessary to detect and prevent such ARP Spoof attacks. Thus ARP poisoning and detection tools have been built and demonstrated via simple tools using scapy which enable the user to attack and detect ARP Spoofing. Prevention of ARP Attacks using static entries is a simple and feasible solution especially if the communication between the two hosts occur often and is employed by us in this project.

Innumerable enhancements and various new advancements can be made to this procedure, a few substeps may be added to make the process more efficient. This procedure need not just be used for ARP attacks but can also be used for other attacks for which more rules may be needed and better tools can be created involving machine learning models especially today as machine learning is one of the top research fields in the branch of computer science. More research can be done for efficient detection and prevention for ARP Poisoning and other such MITM attacks. Use of Dynamic ARP Inspection integrated with DHCP server and Packet Filtering using tools such as Wireshark can also be done for prevention of ARP Spoofing attacks

References
[1] Arote P and Arya K V Jan. 2015 Int. Conf. on Computational Intelligence and Networks, p 12-13
[2] Tripathi N and Mehtre B Nov 2013 An ICMP based secondary cache approach for the detection and prevention of ARP poisoning in Conf. Computational Intelligence and Computing Research (ICCIC), 2013 IEEE Int. Conf.
[3] Bruschi D, Ornaghi A, Rosti E Dec 2013 19th Annual Computer Security Applications Conf., 2013.
[4] Xia J, Cai Z, Hu G and Xu M Jun 2019 Chinese J. of Electronics 28, Pg 172-178
[5] Data M Nov 2018 The Defense Against ARP Spoofing Attack Using Semi-Static ARP Cache Table in Int. Conf. on Sustainable Information Engineering and Technology (SIET)
[6] Jinhua G and Kejian X Jan 2013 Int. Conf. on Computer Communication and Informatics
[7] Pandey P May 2013 3rd IEEE Int. Advance Computing Conf. (IACC), pp. 147-153
[8] Puangprunpitag S and Masusai N May 2009 6th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology
[9] Kumar S, Tapaswi S July 2012 A centralized detection and prevention technique against ARP poisoning in Int. Conf. on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec) 2012 P: 259-264
[10] Nam S Y, Kim D and Kim J February 2010 Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks in IEEE Communications Letters (Volume: 14, Issue: 2, February 2010)
[11] Barbhuiya, F A, Biswas S, Hubballi N, Nandi S July 2011 A host based DES approach for detecting ARP spoofing in IEEE Symposium on Computational Intelligence in Cyber Security (CICS), 2011. pp. 114-121
[12] T. Chomsiri Nov 2018 Sniffing packets on LAN without ARP spoofing Convergence and Hybrid Information Technology, 2008. ICCIT’08. Third International Conference on, vol. 2. IEEE, 2008, pp. 472-477
[13] Lootah W, Enck W and McDaniel P Jan 2006 Tarp: Ticket based address resolution protocol 51 no. 15. Elsevier 2007. pp. 4322-4337
[14] Nayak G N and Samaddar S G Jan 2010 Computer Science and Information Technology (ICCSIT) 3rd IEEE Int. Conf., vol. 5. pp. 491–495
[15] Hussain M A, Jin H, Hussien Z A, Abduljabbar Z A, Abdal S H and Ibrahim A April 2016 International Conference on Network and Information Systems for Computers (ICMISC)
[16] Yang M, Wang Y and Ding H Dec 2014 International Conference on Instrumentation and
Measurement, Computer, Communication and Control

[17] Prevelakis V and Adi W Sept 2017 LS-ARP: A lightweight and secure ARP in 7th International Conference on Emerging Security Technologies (EST)

[18] Alharbi T, Durando D, Pakzad F and Portmann M Nov 2016 Securing ARP in Software Defined Networks in IEEE 41st Conf. on Local Computer Networks (LCN)

[19] Qian A July 2009 The Automatic Prevention and Control Research of ARP Deception and Implementation in WRI World Congress on Computer Science and Information Engineering