International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

# ARP Spoof Detection System using ICMP Protocol: An Active Approach

Mr. Vinay K. R.[1] M.Tech
Dept of Computer Science & Engineering[1]
Ballari Institute of Technology & Management,
Bellary, India

Mr. T. R. Muhibur Rahman[2] Assoc Prof
Dept of Computer Science & Engineering[2]
Ballari Institute of Technology & Management,
Bellary, India

***Abstract*—Address Resolution Protocol (ARP) is used by computers to map network addresses (IP) to physical addresses (MAC). ARP spoofing is the act of vindictively changing the IP-MAC associations stored in ARP cache of any network host. This paper discusses ARP spoofing attack and some related works about it first. On these bases, the paper proposed an efficient algorithm based on ICMP protocol to detect malicious hosts that are performing ARP spoofing attack. The technique includes collecting and analyzing the ARP packets, and then injecting ICMP echo request packets to probe for malicious host according to its response packets.**

***Keywords—ARP cache; ARP protocol; ARP spoofing; libpcap; dsniff; ICMP protocol***

## I. INTRODUCTION

Address Resolution Protocol (ARP) is a protocol having simple architecture and have been in use since the advent of Open System Interconnection (OSI) network architecture. It's been working at network layer for the important dynamic conversion of network address i.e. Internet Protocol (IP) address to physical address or Media Access Control (MAC) address. When a host wants to communicate with another host whose hardware address it does not know, it broadcasts an ARP request for the hardware address associated with the protocol address of the destination. And only the host with corresponding protocol address sends a unicast reply to the sender with its < protocol address, hardware address,> pair. Obviously, ARP protocol plays a key role in local area network communication, but due to its own loopholes, it is often used as part of other serious attacks such as Man-in-the-Middle (MiM) attack, Denial of Service (DoS) attack. With a MiM attack, the attacker can sniff the traffic between two victim hosts. With a DoS attack, the attacker makes a victim host deny communicating with others. So ARP spoofing attack is becoming the most dangerous attack in the LAN [1].

There are two techniques for detecting ARP spoofing one is Passive technique and other is Active technique. In Passive Detection we sniff the ARP requests/responses on the network and construct a MAC address to IP address mapping database. If we notice a change in any of these mappings in future ARP traffic then we raise an alarm and conclude that an ARP spoofing attack is underway. The main drawback of the passive method is a time lag between learning the address mappings and subsequent attack detection. In active method we are purposely injecting the packet to the network. The proposed technique actively interacts with the network to gauge the presence of ARP spoofing attacks. ARP protocol is vulnerable to many different kinds of attacks. These attacks lead to loss of important information. With certain loopholes it has become easy to be attacked and with not so reliable security mechanism, confidentiality of data is being compromised.

The rest of the paper is organized as follows. In section 2 ARP spoofing attack and several serious ARP attacks are described. Section 3 provides an overview of currently available techniques to deal with ARP attacks. Section 4 discusses the proposed method. Section 5 describes the algorithm and finally in section 6 conclusions are made.



## II. BACKGROUND

The most commonly employed network frame architecture is based on OSI model. Each layers service depends on the varying protocols working at layers. But working of these protocols to ensure complete security to the applications running has really become a challenging task for the developers and security professionals. There have been different efforts made in the field of network security to provide effective and reliable defensive mechanisms to control attacks being made of these protocols due to their lack of ability to work and prevent themselves in complex and insecure network. One such protocol is ARP working on network layer. Due to its architecture it cannot prevent itself from being attacked over LAN.

In order not to make the same request in the near future, every host maintains a table called ARP cache to store the address pairs learnt from the network. The host who issued the request will cache the address pair taken by the reply. There are probably two types of entries in an ARP cache: Static entries which remain in the ARP cache until the system reboots Dynamic entries which only remain in the ARP cache for few minutes. Most operation systems allow creating a new entry by an ARP reply packet. All operation systems allow update an old entry by an ARP request or reply packet [2].

As ARP is a stateless protocol and its reply packets are not authenticated, all hosts blindly cache the ARP replies they

receive from the network. This mechanism provides convenient for malicious host. Attackers send forging ARP packets to the victim periodically to perform ARP spoofing attack. In an ARP spoofing attack, the attacker sends ARP request or reply packets with fake <IP, MAC> mappings. For example, if a malicious host wants to sniff traffic sent from X to Y, he could send X an ARP packet with the address mapping <IP of Y, MAC of attacker>. Host X will cache the wrong address mapping and send data destined to Y to the attacker instead. If the attacker also wants to know the information sent from Y to X (MiM attack), the attacker just needs to send Y an ARP packet with address mapping <IP of X, MAC of attacker>. In order not to interrupt normal communication between host X and Y, the attacker need to enable IP packet routing to redirect the packet to the original destination host. If the attacker wants to perform a DoS attack, the attacker can poison the ARP cache of a host in the same way. Every packet the host sends is sent to the attacker. Once the attacker receives the packet, he simply drops it, and therefore, blocks the communication of the victim host [3].

## III. RELATED WORKS

Many efforts have been made and different methods have also been applied to prevent ARP spoofing attacks, but none has been able to give satisfactory results. Different tools and architectures have been proposed but each have their own feasibility issues.

One simple but effective way to prevent ARP attacks is using static entries in the ARP cache. The drawbacks of this solution are its low scalability. It does not work well in dynamic environment and it would be a really heavy work for the network administrator to deploy and update these tables throughout the network especially when the network is big.

Gouda and C.-T. Huang [4] proposed the architecture to resolve IP addresses into MAC addresses over an Ethernet. The proposed architecture consists of a secure server connected to the Ethernet and two protocols: an invite–accept protocol and a request–reply protocol. Each computer connected to the Ethernet can use the invite–accept protocol to periodically record its IP address and its hardware address in the database of the secure server. Each computer can later use the request–reply protocol to obtain the hardware address of any other computer connected to the Ethernet from the database of the secure server. This solution is not practical because it requires changing the ARP protocol implementation of every host with this new address resolution protocol.

D. Bruschi, A. Ornaghi, and E. Rosti. Secure ARP protocol (S-ARP) [5] is a backward compatible extension to ARP. Beginning with an ARP request, Cryptographic Link Layer (CLL) applies public key cryptography to perform an initial handshake between two hosts with the aim to establish a security association. The two hosts prove their identity to each other and exchange keying material. Here upon, secured IP data packets may be sent. This solution involves cryptography to authenticate the origin of ARP packets. To implement this solution in a LAN, every host has to be modified to use S-ARP instead of ARP. This is not scalable to update a stack across all available operating systems. Another disadvantage of this method is that it has the additional overhead of cryptographic calculations as S-ARP uses Digital Signature Algorithm (DSA).

M.Barnaba, Anticap [6] is a kernel patch for UNIX-based operating systems. It prevents ARP poisoning attacks by rejecting ARP updates that contain a different MAC address from the current table entry for the same IP address. This solution works only in static environment, and is available for a limited number of operating systems.

Some high-end Cisco switches have a new feature which allows the switch to drop ARP packets with invalid <IP, MAC)>address bindings [7]. One disadvantage of this feature is its high cost; another one is that it might not be able to verify some ARP packets on all switches in the VLAN.

## IV. THE PROPOSED ARCHITECTURE FOR DETECTING ARP SPOOFING

The following sections describe the details of the architecture we propose and tools which are used to achieve the task.

### A. *The composition of the experiment is listed here:*

- Libpcap used as a Packet capture tool.
- Dsniff used as a spoofing tool.

Libpcap is an open source library that provides a high level interface to network packet capture systems. Libpcap runs on most UNIX-like operating systems (Linux, Solaris). There is also a Windows version named Winpcap. Today, libpcap is maintained by the Tcpdump Group [8].

Dsniff is a collection of tools for network auditing and penetration testing [9].

In this paper, we mainly use two types of packet: ARP packet and ICMP ping packet. The headers are as follows:

```
/*structure of Ethernet header*/
        struct ETHER_HEADER
        {
                u_char dmac[6];
                u_char smac[6];
                u_short type;
        };
/*structure of ARP header*/
        struct ARP_HEADER
        {
                unsigned short arp_hdw_type;
                unsigned short arp_pro_type;
                unsigned char arp_mac_len;
                unsigned char arp_pro_len;
                unsigned short arp_opt_type;
                unsigned char arp_src_mac[6];
                unsigned char arp_src_ip[4];
                unsigned char arp_dst_mac[6];
                unsigned char arp_dst_ip[4];
        }
/*structure of ICMP header*/
        struct ICMP_HEADER
        {
                byte i_type;
                byte i_code;
                ushort i_cksum;
                ushort i_id;
                ushort i_seq;
        };
```

### B. Architecture

As shown in Fig. 1, we adopt a modularized approach and divide our ARP spoofing detection system into the following modules:

- **ARP Packet Sniffer Module:** This module sniffs all ARP packets from the Ethernet.
- **IP-MAC Mapping Database:** Initially first IP-MAC entry is stored in this database and second entry is compared with database, If both entry matches it indicates no sign of ARP spoofing but if new MAC entry for same IP entry it assumed to be spoofing hence we send that IP entry to ARP spoofing detection module, and if new IP entry comes then it stored in database.
- **ARP Spoofing Detection Module:** This is the main Detection module. We feed the IP entry for which new MAC entry came into it as input. This module send ICMP packet to that IP address and if reply comes from that host then it decides the host is legitimate else if reply doesn't come then decides the host is fake, with reply it obtains real MAC and the database is updated with real MAC. The details of the module will be discussed in Section C.
- **Response Module:** This module is used to alert the network administrator the happening of ARP spoofing attack

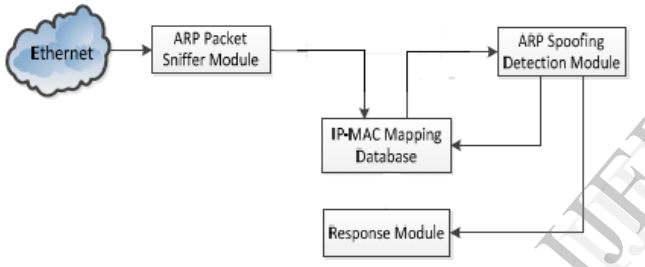


Fig. 1. ARP Spoofing detection architecture.

### C. ARP Spoofing Detection Module

It works based on two rules as follows.

**Rule 1:** The NIC of a host only accepts packets with its own hardware address, broadcast address, and subscribed multicast addresses. The network layer only accepts IP packets addressed to its IP address and will drop the other packets silently. For example, there is a host with MAC address X and IP address Y, it would accept packet with destination MAC address X and destination IP address Z as the destination MAC address matches, but still discard the packet as the destination IP address doesn't match, without sending any error messages back to the source host.

**Rule 2:** Hosts with enabled IP packet routing will forward the packet to the destination host. All legitimate hosts in the network do not enable IP packet routing and will response back after it receives an ICMP echo request packet.

Based on the two rules, we can verify the ARP packets we've got whether they are real or fake packets.

## V. The Detection Algorithm For ARP Spoofing

```
1: If packet type is 0X806 then ARP packet.
2: Store First IP-MAC entry into database.
3: If (stored first entry == next obtained entry) {
        Capture next packet}
   Else if (only IP matches) {
        Send IP to ARP spoofing detection module}
   Else (new IP entry) {
        Store into database}
4: ARP spoofing detection ( ) {
        Send ICMP packet to obtained IP
        If (reply comes) {
                Host is legitimate}
        Else
                Malicious host and call response module ( )
    }
5: Response module ( ) {
        Alert message to network administrator
   }
```

## VI. Conclusions

In this paper we study the theory of ARP spoofing attack and various existing techniques proposed to defend against this attack, and then, we proposed a comprehensive method to deal with ARP spoofing problem.

As the method we proposed to probe the authenticity of every ARP packet is very active, the time delay between capturing the packets and detecting spoofing attack is minimum. We send one trap ICMP ping packet for each spoofed ARP packet on the network and then infer its authenticity according to the response to our packet.

Our method can also detect correct IP-MAC address mappings of both the true host and the malicious host during an actual attack. This technique is simple and efficient.

## REFERENCES

[1] Stevens, TCP/IP Illustrated: vol. 1 (2001).
[2] Zouheir Trabelsi and Khaled Shuaib, "Spoofed ARP Packets Detection in Switched LAN Networks". pp. 81–91, 2008.
[3] T. Demuth and A. Leitner, "ARP spoofing and poisoning: Traffic tricks" Linux Magazine, 56: 26–31, July 2005.
[4] M. Gouda and C.-T. Huang, "A secure address resolution protocol", Computer Networks, 41(1):57–71, Jan 2003.
[5] D Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: A secure address resolution protocol", In Proceedings of the19th Annual Computer Security Applications Conference (ACSAC '03), Dec. 2003.
[6] M.Barnaba.anticap, <http://www.antifork.org/viewcvs/trunk/anticap>.
[7] Cisco Systems. Configuring Dynamic ARP Inspection, chapter 39, pages 39:1–39:22, 2006.
[8] Luis Martin Garcia, "Programming with Libpcap", 2008.
[9] http://images.ihackmyi.com/news/steve/hacks/dsniff_faq.pdf