Feasibility analysis of different methods for prevention against ARP spoofing

Mr Sumit Miglani , Inderjeet Kaur

Computer Science and Engineering Department, Thapar University, India

Abstract- Address resolution protocol is one of the most critical protocol serving in the OSI model of network architecture. It is responsible for the conversion of network address to physical address at the network layer. But, it is vulnerable to certain attacks and hence information integrity also gets compromised to great extent. Many efforts have been made and different methods have also been applied to prevent such attacks at ARP, but none has been able to give satisfactory results. So, an analysis of such method in order to prevent ARP has been done to layout the feasibility considering different factors like backward compatibility, cost, efficiency, ease of implementation, size of network, reliability, manageability etc. of these respectively.

Index Terms- ARP, MAC address, ARP poisoning, Spoof detection, Port security.

I. INTRODUCTION

A) ARP (Address Resolution Protocol)

The task of determining the MAC(Media Access Control) address for the data to be sent on network is the responsibility of ARP. ARP is used by the IP network layer to map IP addresses to hardware addresses at data link layer. ARP is working below the network layer as a part of the Open Systems Interconnection (OSI) link layer, and is used when IP is used over the Ethernet.

B) How does ARP works?

When an Ethernet frame is broadcasted from one machine to another on LAN, the 48-bit MAC address is used to determine the interface for which the frame is meant to be destined. *Address resolution refers to the process of dynamically finding a MAC address of a computer on a network.* The protocol provides a dynamic mapping between the two different types of addresses that are IP address and MAC address which is used by data link layer. The process is dynamic since it happens automatically and is normally not a concern of either the application user or the system administrator. In a shared Ethernet where hosts use the TCP/IP suite for communication, IP packets need to be encapsulated in Ethernet frames before they can be transmitted on to the wire.

There is a one-to-one mapping between the set of IP addresses and the set of Ethernet addresses. Before the packet can be encapsulated in an Ethernet frame, the host sending the packet needs the recipient's MAC address. Therefore, ARP is used to find the destination MAC address using the IP address.

C) How ARP is compromised?

ARP does not maintain the states of its own and hence does not check whether the upcoming arp reply was actually requested or not, before updating the corresponding pairing in the arp cache of the system. So, the attacker sends the bogus replies to the communicating systems, thereby making the changes favorable to attacker, in the pairing of IP and MAC addresses. By doing this the information starts going through the attacker's machine, without coming into notice of actual hosts.

II. ARP Poisoning

In order to minimize the number of ARP requests that are being broadcast, operating systems maintain a cache of ARP replies from different hosts. When a host receives any ARP reply, it will normally update its ARP cache with the new IP/MAC association entry. Since ARP is known to be stateless protocol, most operating systems generally will update their cache if a reply is received, regardless of fact whether they sent out any actual request or not.

ARP spoofing is mainly construction of forged ARP replies. When a forged ARP reply is sent, a target computer could be easily pursued to send frames meant for Host A to instead go to Host B. If done properly, Host A will have no idea that any such redirecting of data has taken place. *The process of updating a target computer's ARP cache with a forged entry is referred to as "poisoning".* The result of ARP cache poisoning is that the IP traffic intended for one host is diverted to a different host.

There are many different kind of attacks that could be implemented to poison the respective arp caches of two communicating devices. These are like man-in-the-middle attack, sniffing, cloning, connection hijack, denial of service, smart IP spoofing etc. Encrypted connections are also not secure. Such attacks can also be performed on SSL(Secure Socket Layer) also. It has also become easy due to easy availability of different exploits online and that too are free of cost.

III. FEASIBILITY OF PREVENTION METHODS

Every network that can be considered a LAN is exposed to this type attack, no matter what kind of networking technology has been used. There is no universal defense measure or cure against ARP spoofing. But still there are certain preventive measures that could be taken, but they their respective limitations due to certain factors like backward compatibility, cost, efficiency, ease of implementation, size of network, reliability, manageability etc. They still are effective depending upon the network type and security related to data being communicated. Different available techniques could be broadly classified on basis of approaching levels. They are:

1. System level

a) Static IP: Here IP and Mac association is entered in cache by the administrator itself. Therefore forged replies are not able to manipulate the cache. But it also increases the workload of administrator. But for a small network, one is able to protect its gateway effectively.
b) Operating system: In linux, its kernel 2.4 does not respond to the unrequested replies, but updates on requests. And it could be made to respond using tools like ettercap. Solaris[2] also updates its entries after some predefined time bounds. This also does not prevent attacker, since it can manage to reply before the legitimate user working fast enough to meet time limits.
c) Firewalls: These also apply the act of detecting only the modified log entries. If found anything suspicious, intimidate the administrator.
d) Ebtable: It is the utility available in linux for programming the switches. It could be used to avoid ARP poisoning also but much of the task is left on the shoulders of administrator, that one could easily made mistakes while programming. Also these rules for ARP prevention are not widely available.

2. *Hardware level*
   a) Sniffer: Efforts were made by M.M. Dessouky [9] to built a hardware easy to be installed acting as a sniffer for detecting the attacks. But major question is to prevent the attacks because once the attack has been made it gives enough time to the attacker to do malicious task in network.
   b) Port security: What it mainly does, binds a specific MAC address to the port. Performance level is also maintained, requires certain rules to be configured. It provides security from only certain type ARP attacks not all.
   c) Dynamic ARP Inspection[11]: These Cisco's high end switches that update IP/MAC binding after analyzing DHCP IP releases. And effectively drops the invalid replies. But these are quite expensive.
3. *Middleware level*
   a) ArpWatch: Is easily available tool that act as a monitoring agent for the arp related activities. It reads the previous and updated data and alarms the administrator if anything does not match. But it gives a lot of false notifications while in an environment where DHCP is used.
   b) ARP-Guard: It works within architecture employing sensors. A better approach than arpwatch but not good enough.
   c) Snort: It's a kind of intrusion detection system. It constantly observes the network for malicious activity regarding ARP and timely send the information to the administrator. But it is mainly deployed at network borders, and is not worth of deploying within the internal network. This whole approach goes in vain when many IDS system does not consider working with DHCP and not much backward compatible with general ARP.
   d) Anticap: Is a kernel patch that rejects the invalid combination of IP and MAC addresses learning from earlier entries. Does go well with DHCP. Being kernel patch requires kernel space thereby slowing down the performance.
4. *Cryptographic level*
   a) S-ARP: Secure ARP[8] involves cryptography to get the arp replies be digitally signed before they are considered valid for updating the arp cache. It is backward compatible also but requires a signing authority server that will keep track of all the public and private keys of all the participating hosts in the network. It adds to the complexity of the whole network and also slows down the performance as it requires time for validating the digital signatures. In case the authority server fails, it leads to the failure of the whole network.
   b) IP-sec: Secure arp provides authentication at link layer only where as IP-sec can provide more protection with almost the similar overheads. But it puts a lot of load on CPU thereby effecting the performance whereas S-arp leads to less load on CPU.
5. *Architectures proposed:*
   a) A simple approach is to divide the large network into small networks so they are easily maintained by the administrator. In smaller number of hosts any malicious host is also easily identifiable.
   b) Middleware approach given by Tripunitara[12] is not practically implemented. Its is a kind of asynchronous prevention and detection scheme. It requires a lot of changes to be made on all the hosts in the network. Though it is backward compatible but no widely acceptable implementations are available.
   c) Gouda, [13] proposed an architecture using a secure server. The communication with server is done using two protocols. But it is not practical, as it requires new protocols to be installed at every host and failure of secure server will collapse the whole network.

IV. CONCLUSION

Today's techniques can't give you complete protection against ARP attacks, but we can guard our self with IDS and specialized ARP manipulation sensors to detect most manipulation attempts. Ignoring the issue is not a convincing option unless we can genuinely trust every user with access to our LAN. ARP spoofing is one of several vulnerabilities which exist in modern networking protocols, which allow a knowledgeable individual free reign over a network. As we have seen these attacks are relatively easy to implement, as there is a large variety of automated tools available, while any type of defense against them would not be enough.

V. FUTURE SCOPE

As we have seen that there no much reliable and effective technique to prevent from ARP spoofing. So, there still need of a lot of work that could be done. There are many tools available to perform the attack but none to ensure complete security from such attacks. We could purpose some changes in the existing algorithms for ARP Cache poisoning prevention and detection for a host running Linux.

REFERENCES

[1]. Raul Siles, "Real world ARP spoofing", GIAC Certified Incident Handler (GCIH) Practical, Version 2.1a, august 2003.

[2]. Thomas Demuth, Achim Leitner, "ARP spoofing and poisoning", Linux magazine, issue 56, pp. 26-31, July 2005.

[3]. Stephen fewer, "ARP poisoning", Harmony security, research and consultancy.

[4]. Bhirud, S.G., Vijay Katkar, "Light weight approach for IP-ARP spoofing detection and prevention".

[5]. Cristina L. Abad, Rafael I. Bonilla, "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks", 27th International Conference on Distributed Computing Systems Workshops, 2007.

[6]. Somnuk Puangpronpitag, Narongrit Masusai, "An Efficient and Feasible Solution to ARP Spoof Problem".

[7]. Thawatchai Chomsiri, "Sniffing packets on LAN without ARP spoofing". International Conference on Convergence and Hybrid Information Technology, 2008.

[8]. D. Bruschi, A. Ornaghi, E. Rosti, "S-ARP : a Secure Address Resolution Protocol" in the proceedings of the 19th Annual Computer Security Application Conference, 2003.

[9]. M.M. Dessouky, W. Elkilany, N. Alfishawy, "A Hardware approach for detecting the ARP attack".

[10]. Christoph Mayer, "Securing ARP, an overview of threats and approaches", version 0.2.0.

[11]. Cisco Systems. Configuring Dynamic ARP Inspection, chapter 39.

[12]. M. Tripunitara and P. Dutta. "A middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning."

[13]. M. Gouda and C.-T. Huang. "A secure address resolution protocol", Computer Networks.

AUTHORS

First Author – Mr. Sumit Miglani,Assistant Professor, CSED - Thapar University, smiglani@thapar.edu.

Second Author – Inderjeet kaur, B-tech(CSE), ME(CSE) , Thapar University, jeet.inder03@gmail.com.

Correspondence Author – Inderjeet Kaur, jeet.inder03@gmail.com, jeet_inder03@yahoo.com, 9878233775