ISSN:0975-9646
Satya P Kumar Somayajula et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1465-1469

A New Scheme to Check ARP Spoofing: Prevention of MAN-IN-THE-MIDDLE Attack

Satya P Kumar Somayajula[1], Yella. Mahendra Reddy[1], Hemanth Kuppili[2]
[1]*CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India.*
[2]*CSE Department, Vignan's Institute of Engineering for women,Visakhapatnam, A.P, India*

***Abstract*-The Address Resolution Protocol (ARP) is a computer networking protocol for determining a network host's Link Layer address when only its Internet Layer (IP) or Network Layer address is known. Address Resolution Protocol (ARP) spoofing is a technique used to attack an Ethernet wired or wireless network. It also known as ARP flooding, ARP poisoning or ARP Poison Routing (APR). ARP Spoofing may allow an attacker to sniff data frames on a local area network (LAN), modify the traffic, or stop the traffic altogether. The principle of ARP spoofing is to send fake, or "spoofed", ARP messages to an Ethernet LAN. In this paper, we present a novel approach that is an extension of the original ARP that came into existence in 1982, where we feel that if within a network , when a communicating node identifies the correct Media Access Control (MAC) address from a given IP address through request and response mechanism, it is made to store that in a IP/MAC address mapping table for the entire while that machine is alive. By doing this we can put a check to the MAN-IN-THE-MIDDLE (MITM) attack for that IP address. The proposed scheme goes further by taking into consideration a probability based voting resolution mechanism for new IP addresses based is backward compatible with existing ARP.**

Keywords: **ARP Spoofing, IP/MAC mapping table, man-in-the-middle, voting based prevention**

1. INTRODUCTION

Address Resolution Protocol (ARP) is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP V4, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. The physical machine address is also known as a Media Access Control or MAC address. A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions. Every machine connected to the network has a 48 bit identification number. This is a unique number which is assigned at the manufacturing time of tthe card. But, Internet communications do not directly use this number (since the computers' addresses would need to be changed each time a network interface card is changed) but uses a logical address allocated by the system: the IP address. The physical addresses can be mapped to the logical addresses and vice-versa:the ARP protocol interrogates the network machines to find out their physical address, then creates a lookup table between the logical and physical addresses in a cache memory. When one machine needs to communicate with another, it consults the lookup table. If the requested address is not found in the table, the ARP protocol issues a request over the network. All machines on the network will compare this logical address to theirs. If one of them identifies with this address, the machine will respond to ARP which will store the address pair in the lookup table and communication will then be able to take place.

ARP spoofing, also known as the "Man in the Middle" means that an attacker without affecting the clients' normal internet service can cause to redirect all traffic to pass through him passively first, thereby unauthorisidely accessing all secret information. The attacker also has the opportunity to either modify the packets as they pass through in order the change the information, or in a much serious scenario cause blocking of any traffic, which is known as a DoS (Denial of Service). The basic aim of ARP poisoning is to create fake ARP messages which will map the other IP's to the attackers MAC address in the cache's of the client. For example, let's assume the gateway 10.1.1.12 has a MAC address of 0E:33:FB:G13:G12:11, 10.1.1.22 has a MAC of 00:02:FE:G12:1B:CC, and 10.1.1.33 has a MAC of 00:11:22:33:44:55. If 10.1.1.22 was performing the attack, it would send out ARP messages indicating that 10.1.1.12 and 10.1.1.33 was on MAC 00:02:FE:G12:1B:CC, and therefore all traffic destined for either IP address would be sent to that physical MAC address as that traffic is transported over the network layer. At this stage, it is up to the attacker on 10.1.1.22 whether he forwards 10.1.1.33's traffic on to 10.1.1.12, or whether he prevents it from getting there, or alters it on the way. A Denial of Service could also be performed by sending an ARP message informing the clients of new (but non-existent) MAC address has been assigned to the default gateway.

Several extensions to ARP have been made to resolve the ARP cache poisoning problem like Dynamic ARP Inspection (DAI), S-ARP and Ticket-based ARP (TARP). DAI [1] might prevent ARP poisoning, but this demands manual configuration by network managers which is an added task and it has compatibility issues with that part of the network portion that is incapable of DAI thereby making it vulnerable to attacks.

S-ARP[3] and Ticket-based ARP (TARP)[4] are two well known cryptography-based approaches, but involves high computational costs[4]. And there is a risk that the servers, such as Authoritative Key Distributor (AKD) used in S-ARP and Local Ticket Agent (LTA) in TARP, might be subject to failure. In addition, they frequently need the upgrading of the servers, and incremental deployment is never easy. For example, TARP-enabled or S-ARP-enabled machines might not accept ARP replies from non-TARP/S-ARP nodes.

In this paper we investigate a new mechanism to prevent ARP based MITM attacks while overcoming the limitations of existing approaches. Since we are not using any

cryptographic methods and no central servers are involved, there are no complexity issues and a point of failure problem. We introduce two new concepts, retaining memory for longer times(long term memory) and voting, with the existing ARP to resolve the problem, while strictly adhering to the following: backward compatibility with existing ARP, minimal infrastructure upgrade cost (e.g. no upgrade of Ethernet switches or modification of existing hardware), and incremental deployability. The proposed mechanism is experimentally evaluated in a real time environment.

2. ADDRESS RESOLUTION PROTOCOL - MITM DEFIANT.

The proposed Address Resolution Protocol that is immune to MITM, is called as *M D-ARP* and is based on the following concept. When a Node say 'A' knows the correct IP/MAC address mapping for another Node 'B' with which it wants to exchange communication, and if we make Node A retains this mapping all the while that Node B is active and alive, then ARP spoofing based MITM attack between A and B is impossible.

MD-ARP employs an enhanced IP/MAC mapping table, as well as the ARP cache used in existing ARP to retain IP/MAC mapping for alive machines over longer periods. Three fields, IP, MAC, and Timer, are allocated to each IP address registered in the long-term table. The long-term table assigns a default value for the timer it uses, that is 60 minutes. In order to avoid losing the mapping of ($\mathbf{IP_i}$, $\mathbf{MAC_i}$) for a live host after 60 minutes, we send a series of new ARP request messages for IP_i only to MAC_i through unicasting to check if the MAC_i is still alive. Here, 50 ARP request messages are sent at random intervals with an average of 10 msec. We wait for atleast one ARP reply from MAC_i to return and it does then the mapping is registered in the short-term ARP cache and the corresponding long-term table timer is reset to 60 minutes. If no ARP reply returns, then the mapping of ($\mathbf{IP_i}$, $\mathbf{MAC_i}$) is assumed to be invalid and the corresponding entry is deleted from the long-term table. Thus, the IP/MAC mapping can be retained in the long-term table during the entire life of that MAC_i until the binding is released.

The enhanced ARP, MD-ARP attempts to manage the IP/MAC mappings for all active machines in the same LAN by making use of a long-term table. This can be achieved if we fill the long-term table based on the received ARP request messages, in essence the source IP and MAC address fields, since the active machines tend to send ARP requests to find the MAC address of the gateway router repeatedly because of the preset timer value in the ARP cache. However, IP/MAC mapping of every ARP request cannot be directly reflected because of the possibility of ARP cache spoofing. Fig. 1 shows the detailed short-term cache and long-term table update policies that are used on the arrival of ARP request packets or reply packets. By the rule for case (i) in Fig. 1, each IP/MAC mapping registered in the short-term cache is frozen and held constant till it expires, e.g. for 2 minutes for Windows XP, to prevent too frequent cache updates by ARP request sniffing.

In case (ii) of Fig. 1, MAC conflict occurs because the newly received MAC address $\mathbf{MAC_S}$ for $\mathbf{IP_S}$ is different from original MAC_S, say MAC'_S that is already associated with IP_S. The conflict is resolved by giving a priority to original MAC'_S only if it is alive. As shown in Fig. 1, the activity of a host is examined by sending 50 ARP request packets and counting the ARP replies. Multiple ARP requests are sent to avoid unexpected packet losses (the case of Denial of Service) attack on MAC'_S. Even though the packet loss probability is as high as 90% by DoS attack, at least one ARP reply will be returned with a probability of $(1 - 0.9^{50}) = 99.5\%$.

```
/* (IPS, MACS): the sender protocol (IP) and hardware (MAC) addresses of the received ARP request or reply packet */
    if (IPS is registered in the short-term cache)
        {— (i) no action; /* in case of conflict; preserve existing mapping.*/ }

    else if ((IPS, MACS) is registered in the long-term table)
        { register (IPS, MACS) in the short-term cache; set the long-term table timer to 60 minutes ;}

    else if (IPS is in the long-term table, but the registered MAC is not MACS)
        { — (ii) ; /* conflict on IP and MAC mapping. */ }
    send a series of 50 ARP requests to original MAC by unicasting, at random intervals with an avg. of 10 msec;

    if(at least one ARP reply arrives)
        { retain the existing (IP, MAC) mapping, drop the new one; }
    else
        { assign the new mapping; }
          the assigned mapping is registered in the short-term cache, too.}

    else
{ — (iii) send voting requests for IPS; } /* i.e. IPS is not in short-term/long-term tables */

     if (no response)
          the mapping (IPS, MACS) is registered in both tables;
     else if
there exists a MAC that polls over 50% of votes for IPa) that mapping is registered in both tables;}
```

Fig. 1. Short-term cache and long-term table update policies. (Applied on the arrival of ARP request or reply packets)

2.1 Voting-based Resolution Mechanism

Till now, we investigated how to prevent MITM attacks for the nodes whose IP/MAC mapping is known already. How-ever, if Node A receives an new ARP request or reply from a new IP address, then Node A cannot easily judge the authenticity of the received IP and MAC address mapping contained in the ARP message. Consider that a new machine is added in a LAN with no IP/MAC mapping information and the machine sends an ARP request for the gateway router, now if an attacker's ARP reply arrives first, the ARP cache can be easily poisoned or spoofed. In order to solve this problem, we propose a voting-based resolution mechanism which corresponds to case (iii) in Fig. 1.

The basis for the voting-based resolution mechanism is as follows. When Node A is turned on with an empty ARP cache and long-term table, if multiple nearby hosts inform Node A of the true MAC address of the gateway router that they know, then gateway MAC poisoning can be avoided efficiently.

We now find out the details of the voting-based resolution mechanism. Two new ARP packet types are defined for MD-ARP: voting request and voting reply packets. They reuse the packet format of the original ARP request/reply packets. The *operation* field is set to 20 and 21 for voting request and reply packets, respectively. If Node A observes an ARP request or reply from a new IP address IP_R with the MAC address of MAC_R, then Node A broadcasts a voting request with IP_R in its *target protocol address* field to collect IP/MAC mapping for that IP from other neighboring hosts after waiting for a random time of between 0 and 100 msec. The random waiting time is necessary to prevent a simultaneous ARP voting request/reply storming. A neighboring host that receives an ARP voting request for IP_R sends back 50 ARP voting replies with IP/MAC mapping for IP_R at maximum transfer rate without delay provided it is aware of the mapping. Then, Node A calculates the polling score for each received MAC address based on earlier replies. If a MAC exists that received over 0.5 votes, then that MAC address is accepted for IP_R. In order to avoid the bias by the machines with small Round-trip delay time (RTT), we counting only after waiting at least RTT of the machine with the largest RTT in the LAN. Currently, the waiting time before counting is set to 0.3 msec.

When a new MD-ARP enabled machine is added to some LAN and if there are no other such machines, then this new machine cannot benefit from the voting mechanism. However, we can show that this new machine can be additionally protected by voting, if at least two MD-ARP enabled machines exist when there is one attacker. Let us consider a case where k MD-ARP enabled machines $MAC_1, \ldots, MAC_k$ and one adversary MAC $_v$ are interconnected by the same Ethernet switch. When a new MD-ARP enabled machine MAC_e is attached to the same switch, if MAC_e receives an ARP request or reply with a false IP/MAC mapping from the adversary, then MAC_e will broadcast the ARP voting request. Let r_i denote the ARP voting reply traffic rate of MAC_i for i = 1, 2. . . k, and r_v denotes the ARP voting reply rate of an adversary. If MAC_e observes the voting replies during an interval of length l , then the average ratio of voting replies from the adversary becomes $r_v/(r_v + \sum_{i \le k} r_i)$ under the assumption that the Ethernet switch serves input buffers fairly. If the voting reply rate is the same for every machine, then the ratio becomes 1/ (k +1). Since latest machines can send traffic up to near the link rate, the effect of r_v on the ratio of the votes for the adversary is bound to be limited.

We now determine the value of N, number of voting reply packets that is essential to prevent ARP poisoning, when there are two active MD-ARP enabled machines and one adversary node. In this case, the reply ratio from the adversary will be close to one-third by as shown previously. Let us assume that each packets' arrival is independent of other arrivals, and the probability that each packet arrival is from the adversary node is p . X is a random variable that represents the total number of packets from the adversary among N voting reply packets, then X has a binomial distribution, i.e. $X \sim \text{Binomial}(N, p)$. Solving the following inequality using Chernoff bounds formulae [5, Corollary 3.1.2]:

$$\Pr(X/N > \eta) \le \exp\{-2N(\eta - p)^2\}. \quad (1)$$

Since X/N is the ratio of the adversary's replies, if we set η to the decision threshold 0.5, then (1) gives an upper bound on the false negative probability for the adversary. When p= 1/3, if we set N to 120, then the upper bound equals 0.0013. Thus, the value of 120 for N gives a low false negative probability.

Let us investigate the load on the traffic for running the voting-based resolution mechanism. Let L represent the total number of alive machines and and M represent the total number of alive MDefiant-ARP enabled machines in the LAN, respectively. To simplify the analysis, we assume that IP addresses are allocated through Dynamic Host Configuration Protocol (DHCP) for all nodes, and each IP lease time is exponentially distributed with the same average T_D. The MDt-ARP enabled Node A performs voting-based resolution for an active machine with an IP address IP_R only once during its own IP lease time. This occurs when Node A comes up with a newly allocated IP address. When IP_R is released and reallocated, the new mapping is resolved, by the rule for case (ii) in Fig.1, without voting. Thus, the average voting reply rate for IP address IP_B to Node A can be calculated as $(M - 1) \times 50 \times 28 \times 8/T_D = 11.2L(L - 1)/T_D$ Kbps. Thus, the aggregate voting reply rate to Node A is $11.2L(L - 1)/T_D$ Kbps. If M = L , the rate becomes $11.2L(L - 1)/T_D$ Kbps. If we assume that T_D is one day and L = 255, then the average voting reply rate to Node A is about 8.4 Kbps. Thus, the voting traffic overhead is not significant for a small subnet.

3. PERFORMANCE ANALYSIS

We initially check whether a host under DoS attack can respond correctly to ARP requests to check the feasibility of the mechanism corresponding to case (ii) in Fig. 1. We then continue to measure the response probability of a victim host under DoS attack for varying number of ARP requests in a 100 Mbps LAN, and further investigate a single packet response for multiple ARP requests as a successful response of the affected node. For performing the tests we used several DoS attack patterns like SYN flooding, UDP flooding, ICMP flooding, and

ICMP smurf attack. The lowest response probability was obtained from smurf attack because more machines were involved in the attack than for other DoS attack types. The number of involved attack nodes is 25 for smurf attack. Table I shows the result for this attack. We find that the response probability of 99% is achieved if at least 20 ARP packets were sent. Thus, the algorithm for case (ii) in Fig. 1 can work reliably for the specific DoS attack patterns. If a victim node is disabled by another type of DoS attack in the worst case, then the MITM attack cannot be valid by the definition of MITM. However, there is a threat that when the victim recovers, it might be subject to the MITM attack. In this case, if the victim sends a voting request for its own IP address under the assumption that there are a sufficient number of MD-ARP enabled nodes, then the victim can easily know whether its own IP address is used by another machine based on the polling results and avoid MITM attack by stopping the use of the intercepted IP address.

We continue to evaluate the voting-based resolution mechanism in a test environment where six MD-ARP enabled machines, which are 2.66 GHz Dual-core PCs, and several adversary machines, which are 2.66 GHz Quad-core PCs, are interconnected by a Gigabit Ethernet switch. We implemented the proposed mechanism on a Fedora 9 Linux (kernel 2.6.25) by slightly modifying the ARP module code. False decision is made when the average number of votes from the adversaries exceeds N/2.

TABLE I
ARP RESPONSE PROBABILITY FOR VARIOUS NUMBERS OF ARP REQUESTS UNDER ICMP SMURF ATTACK

# of ARP requests	1	5	10	15	20	25
response prob. (%)	24.7	68.4	92.2	97.6	99.6	100

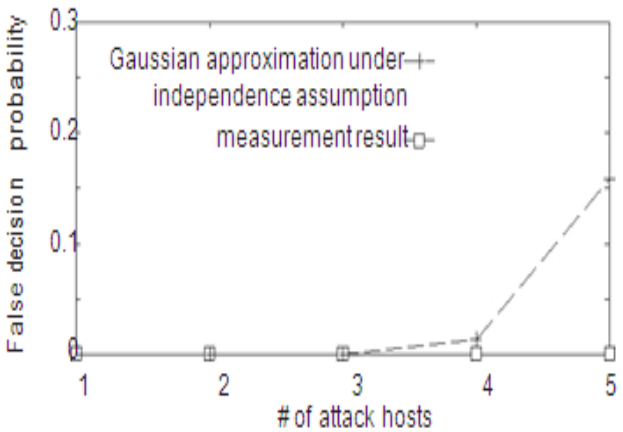

Fig. 2. False decision probability for various numbers of attack hosts when the number of MD- ARP enabled hosts is 6.

The above figure shows the theoretical values vs the measured values of the false decision probability. Each measurement value is obtained from around 100 experiments. The theoretical values are obtained by approximating the binomial distribution of in (1) by Gaussian distribution $\mathcal{N}(N_P , N_P(1 - p))$. We observe that the measured false decision ratios are always zero differently from the theoretical results. When we derive the binomial distribution for X, we assume that the outcome of voting each time is independent from remaining outcomes. However, we found that the votes from different nodes are arriving in an approximately round-robin fashion. Because of this deterministic pattern of voting outcomes, the false decision ratios are measured to be nearly zero when the MD-ARP enabled nodes outnumber the adversary nodes.

Node A that is MD-ARP enabled responds to an ARP request destined for itself by sending an ARP reply message as current ARP. Even though Node A is the only MD-ARP enabled node in the LAN, Node A can accept the received IP/MAC mapping for a new IP address using the rule for (iii) in Fig. 1. This also checks the backward compatibility of MD-ARP with existing ARP.

4. CONCLUSION

A new scheme to prevent ARP spoofing based MITM attacks is proposed and investigated. It relies on two key concepts: long-term IP/MAC mapping table and polling mechanism that were modified and extended versions from the earlier proposed schemes. Though the scheme is installed and tested on a small number of hosts, they can be well protected through voting-based collaboration when extended to large number of hosts. The proposed scheme does not use any kind of cryptography mechanisms and central servers, so it does not have complexity and single point of failure problems while achieving backward compatibility with existing ARP and incremental deployability.

5. FUTURE SCOPE

The current MD-ARP enabled technique can be improved by making a few appropriate enhancements which will result in increased utility of the application in future. Since the MD-ARP is an extension of ARP, its specification (as for message exchange, timeout, cache) follows that of the original one . In order to maintain compatibility with ARP, an additional header is inserted at the end of the protocol standard messages to carry the authentication information. This way, MD-ARP messages can also be processed by hosts that do not implement MD-ARP, although in a secure ARP LAN all hosts should run MD-ARP. Hosts that run the MD-ARP protocol will not accept non-authenticated hosts. But, hosts that run the classic ARP protocol will be able to accept even authenticated messages.

Furthermore, the list of hosts not running MD-ARP must be given to every secured host that has to communicate with an unsecured one.This application made use of standard Ethernet as the medium but can be extended for many newer communication links to obtain comparable results.

In this application we have implemented global threshold values for testing various results, these can be varied for achieving more efficient spoofing prevention technique.

6. REFERENCES

[1] Y. Bhaiji, Network Security Technologies and Solutions. Cisco Press, 2008
[2] I. Teterin, Antidote, http://online.securityfocus.com/archive/1/299929.
[3] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: a secure address resolution protocol," in Proc. Annual Computer Security Applications
[4] W. Lootah, W. Enck, and P. McDaniel, "TARP: ticket-based address resolution protocol," Computer Networks, vol. 51, pp. 4322-4337, Oct. 2007.
[5] S. M. Ross, Probability Models for Computer Science. Harcourt/Academic Press, 2002. Conference (ACSAC), 2003.

Authors Biography

Somayajula Satya Pavan Kumar, working as Asst.Professor, in CSE Department, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India. He has received his M.Sc(Physics) from Andhra University, Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM), Visakhapatnam, A.P., INDIA. His research areas include Image processing, network security and neural networks.

Yella. Mahendra Reddy, received his B.Tech from Gokula Krishna college of Engineering, and he is studying his M.Tech (Software Engineering) second year in the Avanthi Institue of Engineering college & Technology, Tamaram, Visakhapatnam, A.P., India. His research areas of interest are Network security.

Hemanth Kuppili is presently working as Asst.Professor in the department of Computer Science and Engineering at Vignan's Institute of Engineering for Women, Visakhapatnam, Andhra Pradesh, India. He received his M.Tech in Computer Science and Technology from Gandhi Institute of Technology and Management University (GITAM). His areas of research interests are Computer Networks, Information Security and Embedded Systems.