World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:6, No:8, 2012

An Efficient and Secure Solution for the Problems of ARP Cache Poisoning Attacks

Md. Ataullah, Naveen Chauhan

***Abstract*—**The Address Resolution Protocol (ARP) is used by computers to map logical addresses (IP) to physical addresses (MAC). However ARP is an all trusting protocol and is stateless which makes it vulnerable to many ARP cache poisoning attacks such as Man-in-the-Middle (MITM) and Denial of service (DoS) attacks. These flaws result in security breaches thus weakening the appeal of the computer for exchange of sensitive data. In this paper we describe ARP, outline several possible ARP cache poisoning attacks and give the detailed of some attack scenarios in network having both wired and wireless hosts. We have analyzed each of proposed solutions, identify their strengths and limitations. Finally get that no solution offers a feasible solution. Hence, this paper presents an efficient and secure version of ARP that is able to cope up with all these types of attacks and is also a feasible solution. It is a stateful protocol, by storing the information of the Request frame in the ARP cache, to reduce the chances of various types of attacks in ARP. It is more efficient and secure by broadcasting ARP Reply frame in the network and storing related entries in the ARP cache each time when communication take place.

***Keywords*—**ARP cache poisoning, MITM, DoS

I. Introduction

ADDRESS Resolution Protocol (ARP) resides in the Network layer. In a LAN, each computer has a logical (IP) address and a physical (MAC) address. To send a message from one machine to other in the same or different network(s), MAC address of the destination machine is required by the source machine. Therefore to get the MAC address of destination if absent in ARP cache of source, a mapping is needed to be established between the IP address and the MAC address. For this purpose ARP is used. From this it can be understood that ARP is a very important part of the network layer and a stateless protocol. Due to stateless property, ARP have some inherent security flaws which make it vulnerable to different ARP cache poisoning attacks such as MITM and DoS attacks leading to leakage or damage of information.

Due to the importance of this problem, there have been several solutions proposed to solve it. We have analyzed that no solution offers a feasible solution. So in this paper we present an efficient and secure version of ARP that is able to cope up with different types of attacks in ARP and also feasible solution. This modified protocol will retain all of the good points of the original one for ARP [12], but will block off its security weaknesses leading to a more efficient and secured network than existing by making it stateful. We term this modified protocol the "Efficient and Secure Address Resolution Protocol (ES-ARP)". It is a stateful protocol, by storing the information of the Request frame in the ARP cache, to reduce the chances of various types of attacks in ARP.

Md. Ataullah is with the National Institute of Technology, Hamirpur, Himachal Pradesh, INDIA (e-mail: mdataullah@ymail.com).

Naveen Chauhan is with the National Institute of Technology, Hamirpur, Himachal Pradesh, INDIA (e-mail: naveenchauhan.nith@gmail.com).

It is more efficient and secure by broadcasting ARP Reply frame in the network and storing related entries in the ARP cache each time when communication take place.

The rest of this paper is organized as follows. In Section II ARP is described and several possible ARP attacks are outline. Section III gives the detailed of some attack scenarios. Section IV provides a description of the existing solutions to deal with ARP cache poisoning attacks. In Section V, we describe our protocol. The details of algorithms and flowchart are discussed in Section VI. Section VII discusses the comparison with existing solutions. Results are in Section VIII. In Section IX we conclude.

II. Problem Definition

A. Address Resolution Protocol

Address Resolution Protocol (ARP) resides in the bottom half of the Network layer of TCP/IP suite. In this layer, a host is identified by its 32-bit IP address. But the Medium Access Control (MAC) layer of the TCP/IP suite follows a different addressing scheme [6]. An interface in the MAC layer is identified by a 48-bit MAC address. When Network layer receives a packet from the higher layers it checks the IP address of the destination machine. If the destination machine is in the same local network as that of the sending machine, the packet can be sent directly to the destination machine; else the IP packet has to be routed via a router [6]. To send the packet directly to the destination machine, the network layer needs to know the MAC address of the destination machine. The network layer of the TCP/IP suite accomplishes this by using ARP. ARP dynamically maps the 32-bit IP address of a machine to its 48-bit MAC address in a temporary memory space called the ARP cache. There are two types of ARP messages that may be sent by the ARP protocol. One is ARP Request and other is ARP Reply. ARP Request–When a host sends an ARP request, it fills in the ARP Request frame its IP address, MAC address, type of ARP message and the target IP address. Then the ARP request is broadcast to all the hosts in the same LAN as the sending host [6]. The target MAC address field is left blank for the host with the target IP address to fill in. ARP Reply–When a host receives an ARP request containing its own IP address as the target IP address, it fills its MAC address in the target MAC address field and the operation field set to the opcode of the ARP reply. This packet is directly sent only to the requesting machine, this process is called unicast. When the ARP reply is received by the requesting machine it updates its ARP cache with the requested MAC address.

Example of ARP Request and ARP Reply in ARP is as follows:

1. Machine A wants to send a packet to D, but A only knows IP address of D.
2. Machine A broadcasts ARP Request with IP address of D as shown in Fig. 1.

3. All machines on the local network receive the ARP Request which is broadcast.
4. Machine D replies with its MAC address by unicast of ARP Reply as shown in Fig. 2 and update its ARP cache with MAC of A.
5. Machine A adds MAC address of D to its ARP cache.
6. Now Machine A can delivers packet directly to D.

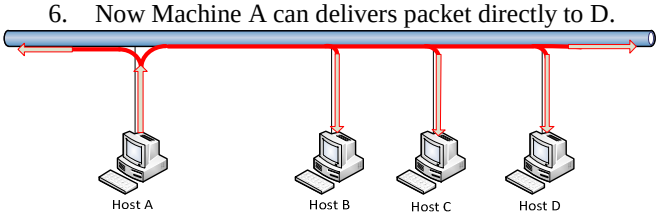

Fig. 1 Host A broadcasts request for Host D

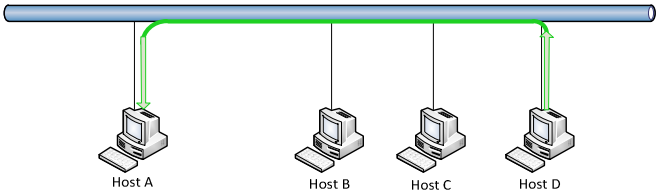

Fig. 2 Host D Replies to Host A (unicast)

Fig. 3 is showing whole process of the ARP operation for more than one network. Source computer A has an address of 172.16.10.100, it is connected to the 172.16.10.0 network, a subnet of 255.255.255.0 is assumed, and we will call this network 1, which is an Ethernet network. Destination computer D has an address of 172.16.20.200, it is connected to the 172.16.20.0 network, a subnet of 255.255.255.0 is assumed, and we will call this network 2 which is also an Ethernet network. Router interface e0 has an address of 172.16.10.99 and router interface e1 has an address of 172.16.20.99. Router interfaces are connected to the 172.16.10.0 (network 1) and to the 172.16.20.0 (network 2) respectively.

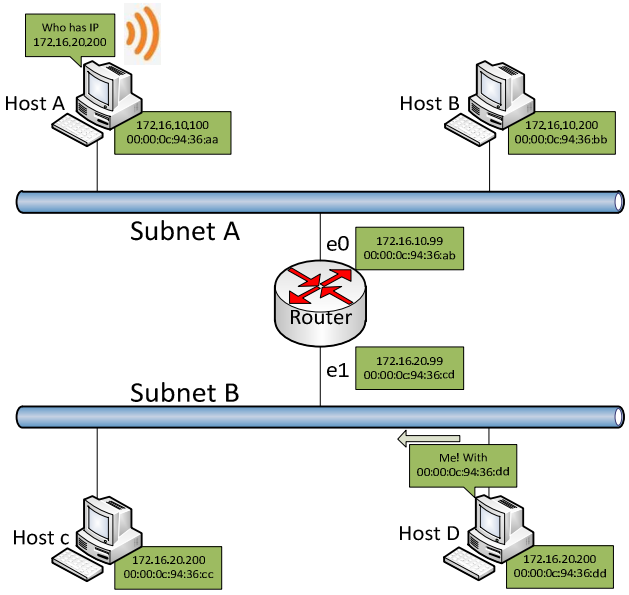

Fig. 3 ARP operation for more than one network

Source host A wants to send some data to destination host D. We will assume that none of the information is stored in the ARP cache on any of the hosts or routers. Source host will create an IP packet addressed to 172.16.20.200. That packet will be sent to the data link layer where it needs a MAC address. Based on the subnet mask, source host will know that the destination host is not on the same local network. So, source host will send out an ARP request for the default router interface's MAC address i.e. what is the MAC for 172.16.10.99. On receiving the MAC address, source host will send out the IP packet (still addressed to 172.16.20.200) encapsulated within a data link frame that is addressed to the MAC address of router interface e0's interface on network 1 (because routers have more than 1 interface they can have more than 1 MAC address, in this case each router has 2 Ethernet interface each with its own unique MAC address).

The routing table will also show the IP address for the next hop is 172.16.20.99. Router interface e0 will forward the frame to router interface e1 by asking for MAC of router interface e1 and it will receive this frame and send the data portion up to the network layer (Layer 3). When router interface e1 receives this frame it will do the same thing that router interface e0 did, it will send the IP packet up to the network layer (the packet is still addressed to 172.16.20.200). The destination host will see that the data link frame is addressed to it and will pass the IP packet to the network layer. At the network layer, the IP address will also match that of the host and the data from the IP packet will be passed up to the transport layer. Each layer will examine the header and determine where to pass it up to until eventually, the data reaches the application running on the destination host that has been waiting for the data.

What is noticed is that through this whole process IP address never changes. The IP packet is always addressed to 172.16.20.200.

Since an ARP gets the message to the target machine, one might wonder why bother with IP addresses in the first place. The reason is that ARP requests are broadcast onto the network, requiring every station in the subnet to process the request.

B. ARP cache poisoning attacks

ARP cache poisoning is the technique by which an attacker maliciously modifies the mapping of an IP address to its corresponding MAC address in the ARP cache of another host [6] by sending spoofed ARP reply. So this technique is also called ARP spoofing.

In Fig. 4 the attacker is Host C. It executes the ARP Cache Poisoning attack by sending a spoofed ARP reply to Host A saying that 'IP address of Host B maps to MAC address of Host C' and a spoofed ARP reply to Host B saying that 'IP address of Host A maps to the MAC address of Host C'. ARP is a stateless protocol and replies are not checked against pending requests. Hence Host A and Host B will update their ARP cache with the mapping received in the ARP replies.

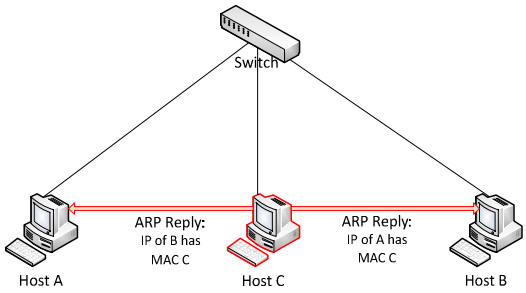

Fig. 4 Host C performing ARP cache poisoning attack on Host A & Host B

C. Man-in-the-Middle (MITM) attack

Once the ARP caches of Host A and Host B are poisoned, Host A will send all the traffic destined for Host B, to Host C. Similarly Host B will send all traffic destined for Host A, to Host C. Host C can now read all the traffic between Host A and Host B. If Host C forwards the packets, after reading them, to the actual destination machine, then Host A and Host B will not even detect that they are being attacked. This is a Man-in-the-Middle attack by which the attacker can divert the traffic passing between two machines to pass via him [6].

In Fig. 5 the attacker is Host C. Host C can divert the traffic passing between two machines to pass via him.

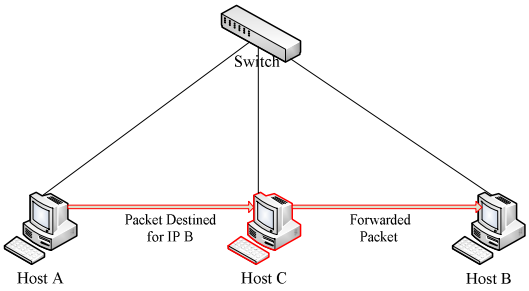

Fig. 5 Man-in-the-middle (MITM) attack

D. Denial-of-Service (DoS) attack

A Denial-of-Service attack is an attempt to make a computer resource unavailable to its intended users. It generally consists of the concerted efforts of a person, or multiple people to prevent the service from functioning efficiently. It is slightly different from MITM attack, when the attacker does not forward the packets, after reading them, to the actual destination machine. This is called Denial-of-Service attack.

In Fig. 6 the attacker is Host C. Host C does not forward the packets, after reading them, to the actual destination machine.

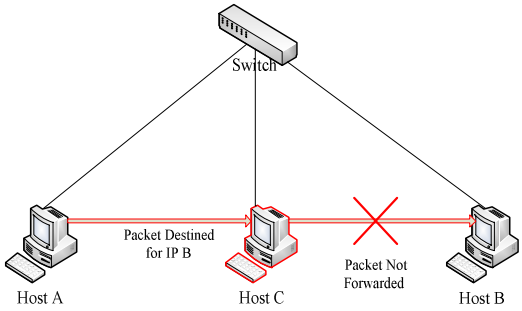

Fig. 6 Denial-of-Service (DoS) attack

III. Attack Scenarios

ARP cache poisoning is an attack having a strong effect in LANs, i.e., all hosts connected to the same switch or hub as that of a malicious host is vulnerable to this attack. Access points act as hubs for wireless networks and act as bridges between wireless networks and wired networks. The detailed of some attack scenarios in network having both wired and wireless hosts are as follows:

A. Attacking wired clients using a wireless client

A wireless attacker can perform a MITM attack against two machines on the wired network connected to the same switch through the access point. In this scenario as shown in Fig. 7, a wireless client, the Attacker, sends a spoofed ARP packet to Host A stating that Host B's IP address is mapped to the Attacker's MAC address. Similarly the Attacker sends a spoofed ARP packet to Host B stating that Host A's IP address has the Attacker's MAC address. Thus the Attacker poisons the ARP caches of Hosts A and B, thereby directing the traffic between them to go through the Attacker [11].

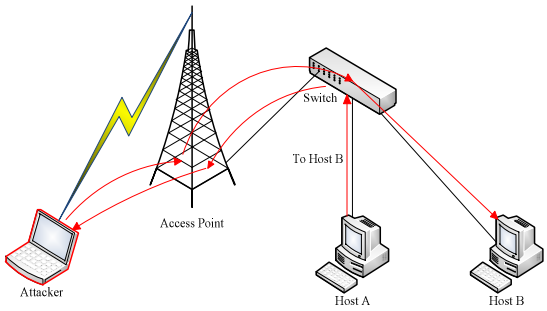

Fig. 7 Wireless client attacking wired clients

B. Attacking a wireless client and a wired client

A wireless attacker can perform a MITM attack against a wireless client connected to a machine on the hub or switch that the access point is connected to. In Fig. 8, the Attacker sends spoofed ARP packets to wired Host B and Wireless Host A, thereby poisoning their ARP caches. Both the victims are in the same broadcast domain as that of the Attacker, hence spoofed ARP packets will reach the victims [11].

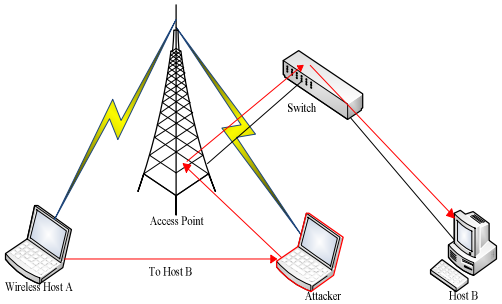

Fig. 8 Wireless client attacking a wired client and a wireless client

C. Attacking wireless hosts

A wireless attacker can perform a MITM attack against two other wireless clients connected to the same access point, as

they are in the same broadcast domain as shown in Fig. 9, a wireless client, the Attacker, sends a spoofed ARP packet to wireless Host A stating that wireless Host B's IP address is mapped to the Attacker's MAC address. Similarly the Attacker sends a spoofed ARP packet to wireless Host B stating that wireless Host A's IP address has the Attacker's MAC address. Thus the Attacker poisons the ARP caches of both wireless Hosts A and B, thereby directing the traffic between them to go through the Attacker. This is a trivial case that is identical to performing an ARP cache poisoning attack in a solely wired environment [11].

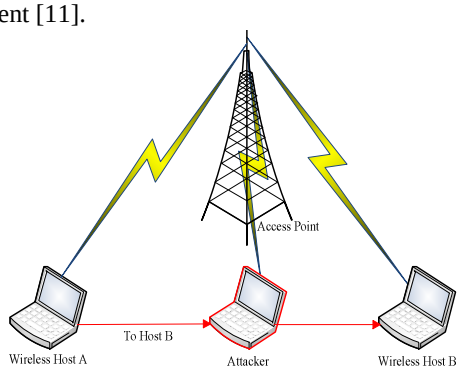

Fig. 9 attacking wireless clients

D.Attacking roaming wireless hosts

A wireless attacker can perform a MITM attack against two wireless clients on different access points (APs) in a roaming setup involving multiple APs. In 802.11b networks, to achieve roaming, the APs need to be connected to the same switch [11] as shown in Fig. 10, there are multiple APs connected to the same switch. Due to this set up all the wireless hosts associated with these APs belong in the same broadcast domain. Hence any forged ARP packet sent from the Attacker can reach any wireless host connected to any of these APs.

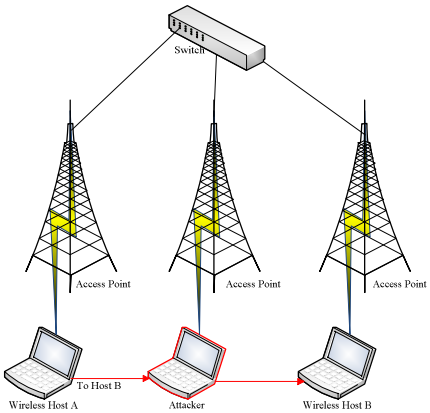

Fig. 10 Attacking roaming wireless hosts

IV. Existing Solutions For Securing ARP

Recently, there have been several existing solutions to solve the ARP cache poisoning attacks problem by adding security to ARP in order to prevent/detect ARP cache poisoning. However, most of them have some critical drawbacks. They are described below in compact form with their strengths and limitations as follows:

A. S-ARP: a Secure Address Resolution Protocol

Bruschi et al. [1] present a secure version of ARP that provides protection against ARP cache poisoning. Each host has a public/private key pair certified by a local trusted party on the LAN, which acts as a Certification Authority. Messages are digitally signed by the sender, thus preventing the injection of spurious and/or spoofed information. It add-on cryptographic features have caused some serious performance penalty and this protocol is hardly in the real world and is not compatible with the standard ARP. So, it is infeasible solution.

The single point of failure is also possible due to failure of Authoritative Key Distributor (AKD) which inserts the public key and the IP address in a local data base, after the network manager's validation.

B. TARP: Ticket-based Address Resolution Protocol

Lootah et al. [2] introduce the Ticket-based Address Resolution Protocol (TARP). TARP implements security by distributing centrally issued secure MAC/IP address mapping attestations called tickets, are given to clients as they join the network and are subsequently distributed through existing ARP messages. Tickets authenticate the association between MAC and IP addresses through statements signed by the Local Ticket Agent (LTA). Each ticket encodes a validity period as an expiration time. Of course, the use of expiration times assumes some form of loose clock synchronization between the issuer LTA and the validating clients. They give a suggestion to change the design of ARP implementation using some cryptographic techniques for creating tickets, have caused some serious performance penalty and this protocol is hardly in the real world and is not compatible with the standard ARP. So, it is infeasible solution.

C. ARP spoofing detection on switched Ethernet networks: A feasibility study

Carnut et al. [3] describe a set of techniques to detect ARP cache poisoning attacks on switched Ethernet networks, both by suggesting implementations to be made directly to the switches' firmwares and alternative techniques that rely only on external elements, such as specialized sniffers and inference from SNMP data collection. It proposed architecture for the detection of ARP spoofing attacks on switched networks. Their architecture requires no special software to be installed on the network hosts. Instead, it delegates the task of detection to one or more detection stations. Their experiments showed that the architecture was very good at detecting ARP attacks without generating false positives. However, it requires a complex setup and attackers could hide behind volume traffic to remain undetected for reasonably long periods.

D.A Hardware Approach for detecting the ARP Attack

Dessouky et al. [4] describe Address Resolution Protocol (ARP) and the ARP cache poisoning attacks and presents a proposed architecture for detecting the ARP attacks. In addition, it discusses a set of techniques used to detect the ARP poisoning attacks on switched Ethernet networks. A new practical technique by adding external hardware element to the LAN network to work as sniffer is suggested. These external

elements are combined in architecture for practical implementation in production network.

It adds external hardware element to the LAN network to work as sniffer is suggested. These external elements are combined in architecture for practical implementation in production network. If working load of sniffing is too high due to increasing in the number of hosts in the network then this external hardware element may fail to sniff the attacks. So it is not a feasible solution of ARP attacks. Cost factor may arise for the users to buy external hardware element.

E. An Efficient and Feasible Solution to ARP Spoof Problem

Puangpronpitag et al. [5] proposed a prototype system, called Dynamic ARP-spoof Protection & Surveillance (DAPS) System; they compare ARP cache poisoning attacks of any IP address as germ infection. They use a valid "static ARP entry" of (IP, MAC) mapping as a "vaccine" to protect against the germ of that IP address. By specifying valid static ARP entries into their ARP cache for all IP addresses that hosts want to communicate, have all needed vaccines against the ARP cache poisoning attacks. They design a prototype system to automate the vaccine provision and injection jobs. This system consists of four components: 1.Gateway Protection (GP), 2.Client Protection (CP), 3.Server Protection (SP), 4.Surveillance Server (SS). This prototype system detects/protects ARP attacks with the help of these four components by using vaccine. It is very complex architecture, so it is hard to manage for the network administrators to manage vaccine in different hosts. It is too expensive to manage the components of DAPS.

F. A middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning

Tripunitara et al. [8] proposed a middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning attacks [7]. Their implementation requires a "Streams based protocol stack," but could be ported to other platforms. The proposed solution is to block unsolicited ARP replies and to raise alarms when a reply is inconsistent with the currently cached ARP entry. Implementing this scheme requires the installation of the middleware on every host on the network. The middleware was also designed to work in the presence of gratuitous ARP messages and proxy ARP servers. One important limitation of this solution is that since it depends on duplicates to detect attacks, it does not prevent/detect attacks in which the host being spoofed is down or being attacked by DoS.

G. A secure address resolution protocol

Gouda et al. [9] proposed architecture for resolving IP addresses into hardware addresses over an Ethernet. The architecture consists of a secure server connected to the network and two protocols used to communicate with the server: an invite-accept protocol and a request-reply protocol. The invite-accept protocol is used by hosts to register their (IP, MAC) mappings with the server. The request-reply protocol is used by hosts to obtain the MAC address of a host connected to the LAN, from the database of the secure server.

This solution is not practical as it requires changing the ARP protocol implementation of every host with this new address resolution protocol [7]. Another disadvantage of this solution is that the secure server represents a single point of failure in the network, and becomes an obvious target for DoS attacks.

V. Efficient and Secure ARP

From previous discussions on ARP it is clear that the main weakness of ARP lies in the fact that it is all-trusting i.e. it does not differentiate between messages received and trusts any received reply blindly. This occurs due to the fact that ARP is a stateless protocol. It does not keep any information regarding the requests it sends out to the network or the replies it receives. This loop hole is used by attackers to send spoofed up replies which are trustingly accepted by the ARP. Thus these replies lead to ARP cache poisoning.

Our implementation of the Efficient and Secured Address Resolution Protocol (ES-ARP) will be in, such a way, that both ARP reply and ARP request is broadcasted. We wish to make ES-ARP stateful by storing the information of the Request frame in the ARP cache. In this protocol all hosts except the source host will store the entries in the ARP cache while the broadcast of both ARP Request and Reply. Example of ARP Request and ARP Reply in ES-ARP is as follows:

1. Machine A wants to send a packet to D, but A only knows IP address of D.
2. Machine A broadcasts ARP Request with IP address of D as shown in Fig. 1.
3. All machines on the local network receive the ARP Request which was broadcasted and update their ARP cache with the MAC of A.
4. Machine D replies with its MAC address by broadcasting ARP Reply as shown in Fig. 11.
5. All machines add the MAC address of D to their ARP cache.
6. Now Machine A can delivers packet directly to D.

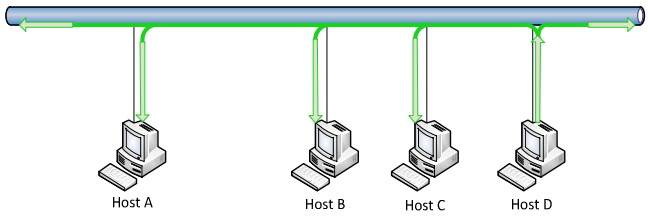

Fig. 11 Host D Replies to Host A (broadcast)

These modifications in the original ARP makes it much more secure as well as efficient.

In the proposed stateful protocol, whenever any ARP reply will arrive to source host, it will check in its ARP cache whether the destination host entry is present or not. If the entry is present, then only the source host will accept the reply, otherwise it will simply discard the ARP reply frame.

This algorithm will check for a valid combination of IP and MAC address of the destination present in the ARP Reply frame, with the created hosts in network(s), before broadcasting Reply frame.

For efficiency, this algorithm broadcasts ARP reply frame too, so updating of the ARP cache will take place twice i.e. first time when ARP request frame is broadcasted (IP and MAC of source host will be stored) and second time when ARP reply frame is broadcasted (IP and MAC of destination host will be stored).

Mathematically:

Let us suppose , if there are N number of hosts in a network(s), then total number of transactions require for the complete updating of ARP cache of all hosts in ES-ARP is given by,

N is even, No. of transaction = N/2.

N is odd, No. of transaction = (N+1)/2.

In case of existing ARP,

No. of transaction = N (N-1)/2.

The broadcasting of ARP reply frame also provides security against ARP cache poisoning, as if any attacker send spoofed ARP reply, then this reply also received by the targeted host whose IP address is used to map with MAC address of attacker. So this host detects that this ARP reply is spoofed by the attacker. Hence we can say the feature of broadcasting of ARP reply frame makes ES-ARP more secure as well as efficient.

VI. Algorithms and Flowchart

The protocol can be shown in detail as follows, Communication from Source Host to Destination Host i.e. for sending ARP request frame and then receiving ARP reply frame.

Procedure ES-ARP Communication (Source → Destination)
BEGIN:
//Before broadcast ARP Request Frame, source host will
//check MAC address of destination host in its ARP cache.
if (ARP cache contains MAC address) *then*
Message will deliver directly to the dest. host
else
Broadcast ARP Request Frame in the channel
//ARP cache will updated by all host except source host.
if (the source network contains the dest. host) *then*
Broadcast the ARP Reply Frame
else
if (not current host)
Update the ARP cache of unmatched host
else
Dest. host is in different network
//Router will be consulted for destination network.
if (routing table contains entry) *then*
Broadcast the ARP Request Frame in dest. n/w.
else
Update the Routing Table by using ARP Frame and recheck
//Destination host is search in destination network.
if (the dest. network contains the dest. host)
then
//Before broadcasting we are checking for any types of attacks
if (IP & MAC of dest. host in ARP Reply frame is valid) *then*
Broadcast the ARP Reply Frame
else
Invalid combination
else
if (not current host)
Update the ARP cache of unmatched host
else
Router will forward the frame to other network
//Check for correct match of source host in the Routing Table.
if (the Routing Table contains correct source network address) *then*
ARP Reply Frame will be broadcasted in the source Network.
//On receiving ARP Reply frame by the source host, it will
//update its ARP cache entry and send message.
if (the MAC address in the frame is same as destination host MAC) *then*
Deliver the message to the destination host
else
Host not found
//As soon as message is delivered, Acknowledgement will be
//send to source host by destination host.
if (the source host match found) *then*
Acknowledge delivered successfully
//So, in this way complete communication will take place.
END: //end of procedure

The flow chart for the ES-ARP protocol is shown in Fig. 12. It depicts the scenario when Source Host wants to communicate to Destination Host and how the protocol works under stateful operations to cope up different types of attacks in ARP which are described in part B of section II.

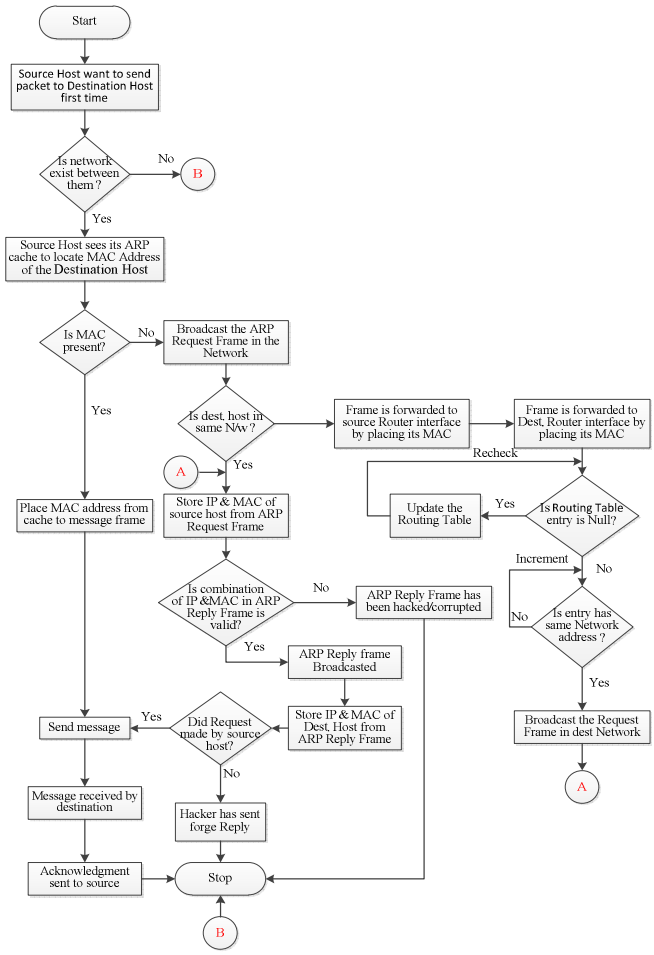

Fig. 12 The flowchart showing the procedure of ES-ARP operation

VII. Comparison with Existing Solutions

Out of the all proposed solutions to solve the problem of ARP cache poisoning attacks, the solutions based on cryptography [1, 2] have caused some serious performance penalty, the single point of failure is also possible and these protocols can hardly exist in the real world and are not compatible with the standard ARP. Proposed by Carnut et al. [3] seems ideal in terms of reducing false positives, but requires a complex setup. Proposed by Dessouky, et al. [4] is more expensive. The middleware approach proposed by Tripunitara et al. [8] is not practical, as it requires changes on all the hosts in the network, and furthermore, no implementation is widely available for download. The prevention/blocking solution proposed by Gouda et al. [9] are the most ambitious ones, but either they require complex installations that do not scale well, are limited to static networks, or require changes on all hosts on the network. On the other hand, proposed by Puangpronpitag et al. [5] is comparatively more effective but increases the work of the network administrator to maintain the different components of DAPS. We have compared ES-ARP with existing solutions as shown in table I..

VIII. Results

Our proposed Stateful protocol ES-ARP retains all of the good points of the ARP but blocks off its security weaknesses. It is also a feasible solution because it does not require any additional host, new device or switches to be added to the network. Cryptography is not used in ES-ARP as in S-ARP [1] and TARP [2], so performance degradation does not occur. Fig. 13 shows the efficiency of ES-ARP.

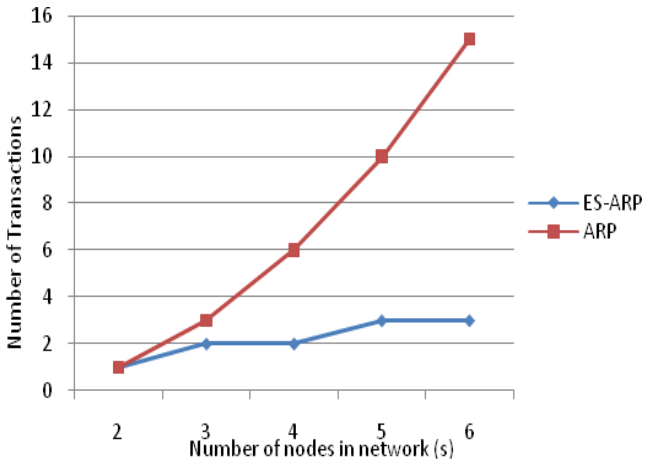

Fig. 13 Efficiency of ES-ARP in terms of transaction

IX. Conclusion

ARP cache poisoning is a serious problem for LAN security. Although there have been several solutions recently proposed to solve the problem, we have analyzed that no solution offers a feasible solution. So, we have proposed an efficient and secure version of ARP that is able to cope up with different types of ARP attacks and is also a feasible solution. ES-ARP is a stateful protocol, by storing the information of the Request frame in the ARP cache, to reduce the chances of various types of attacks in ARP. It is more efficient and secure by broadcasting ARP Reply frame in the network and storing related entries in the ARP cache, each time when communication takes place. It retains all of the good points of the ARP but blocks off its security weaknesses.

TABLE I
SHOWING COMPARISON WITH EXISTING SOLUTIONS

Existing Solution	Crypto-graphy used	Hosts on network	New device added to network	Switches	Performance Degradation	Mechanism
S-ARP [1]	Yes	Trusted Host Authoritative Key Distributor (AKD)	N/A	N/A	High	Signed ARP replies
TARP [2]	Yes	Trusted Host Local Ticket Agent (LTA)	N/A	N/A	Low	Centrally issued tickets authenticate (IP, MAC) associations
Carnut et al. [3]	No	N/A	Special SW required	Port mirroring	No	Sniffing and SNMP heuristics to generate alarms
Dessouky et al. [4]	No	N/A	The HW board is connected to switch	N/A	No	Ping protocol to generate alarm
Puangpronpitag et al. [5]	No	Special Host GP, CP, SP and SS	N/A	N/A	Low	Prototyped System
Tripunitara et al. [8]	No	Special middleware	N/A	N/A	Very Low	Heuristics used to block ARP replies at receiver
Gouda et al. [9]	No	Special Secure Server	N/A	N/A	N/A (replaces ARP)	Secure server resolves queries
Proposed Protocol ES-ARP	No	N/A	N/A	N/A	N/A(modified ARP)	Stateful protocol and broadcasts both ARP request and reply

ACKNOWLEDGMENT

First we would like to thank our Department of Computer Science & Engineering, NIT Hamirpur, which was always there for us listen our problems, give their valuable advices and providing resources for this research. Valuable ideas and feedback on the early draft of this paper were received from Santanu Kumar Sen, Ramesh Kumar and Rajeev Ranjan Patel. We are also grateful to the reviewers for fruitful comments. Last but not least we want to express our sincere thanks to MHRD, Government of India for scholarship.

REFERENCES

[1] D. Bruschi, A. Omaghi and E. Rosti, "S-ARP: a secure address resolution protocol," in Proceedings of the 19th Annual Computer Security Applications Conference, December 2003.
[2] W. Lootah, W. Enck and P. McDaniel, "TARP: Ticket-based address resolution protocol," in Proceedings of the 21st Annual Computer Security Applications Conference, December 2005.
[3] M. A. Carnut and J. C. Gondim, "ARP spoofing detection on switched Ethernet networks: A feasibility study," in Proceedings of the 5th Simpósio Segurança em Informática, November 2003.
[4] M. M. Dessouky, W. Elkilany, and N. Alfishawy, "A Hardware Approach for detecting the ARP Attack," in 7th International Conference on Informatics and Systems (INFOS), May 2010.
[5] S. Puangpronpitag and N. Masusai, "An Efficient and Feasible Solution to ARP Spoof Problem," in 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, May 2009.
[6] Roney Philip, "Securing Wireless Networks from ARP Cache Poisoning," (2007).Master's Projects. Paper 131.
[7] Cristina L. Abad and Rafael I. Bonilla, "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks," in 27th International Conference on Distributed Computing Systems Workshops, 2007.
[8] M. Tripunitara and P. Dutta, "A middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning," in Proceedings of the 15th Annual Computer Security Applications Conference, December 1999.
[9] Mohamed G. Gouda and Chin-Tser Huang, "A secure address resolution protocol" in the International Journal of Computer and Telecommunications Networking, Computer Networks, Elsevier, Volume 41, Issue 1, pages: 57-71, January, 2003.
[10] B. Issac and L. A. Mohammed, "Secure Unicast Address Resolution Protocol (S-UARP) by Extending DHCP," in 13th IEEE International Conference on Networks, 2005. Jointly held with the IEEE 7th Malaysia International Conference on Communication 2005.
[11] B. Fleck and J. Dimov, "Wireless Access Points and ARP Poisoning: Wireless vulnerabilities that expose the wired network,".
[12] D. C. Plummer, "An ethernet address resolution protocol," in RFC 826, 1982.
[13] B. Issac, "Secure ARP and Secure DHCP Protocols to Mitigate Security Attacks," in International Journal of Network Security, Vol.8, No.2, PP.107-118, March, 2009.