A framework to mitigate ARP sniffing attacks by cache poisoning

Article *in* International Journal of Advanced Intelligence Paradigms · January 2018

DOI: 10.1504/IJAIP.2018.10010532

2 authors:

Prabadevi. B
VIT University
55 PUBLICATIONS **1,288** CITATIONS
SEE PROFILE

Jeyanthi Nagamalai
VIT University
95 PUBLICATIONS **525** CITATIONS
SEE PROFILE

A framework to mitigate ARP sniffing attacks by cache poisoning

B. Prabadevi* and N. Jeyanthi

School of Information Technology and Engineering,
VIT University, Vellore, India
Email: prabadevi.b@vit.ac.in
Email: njeyanthi@vit.ac.in
*Corresponding author

Abstract: Today in the digital era of computing, most of the network attacks are caused by sniffing the sensitive data over the network. Among various types of sniffing attacks, ARP sniffing causes most of the LAN attacks (wired and wireless LAN coexist). ARP sniffing causes poisoning of ARP cache or spoofing. Through ARP sniffing, the attacker tries to know the (IP, MAC) pair of victim's system available in ARP table or ARP request-reply packet passed over the network and either exploits victim's resources or creates a situation to deny victim's services for its legitimate users. This in-turn causes MITM, DoS or DDoS attacks. The major cause for these attacks is lack of effective authentication mechanisms with ARP or RARP protocols used for address resolution. This paper provides the working principle of ARP protocol and a method to mitigate the attacks caused by ARP cache poisoning. The proposed framework compares the IP-MAC pair in the ARP and Ethernet headers and if any fake entry is suspected, the information is updated in the fake list and a message is sent to the gateway or router to alert it from cache poisoning attacks.

Keywords: ARP cache poisoning; address resolution; man-in-the-middle attacks; host impersonation; mitigation; ARP sniffing attacks; DDoS attacks.

Reference to this paper should be made as follows: Prabadevi, B. and Jeyanthi, N. (2018) 'A framework to mitigate ARP sniffing attacks by cache poisoning', *Int. J. Advanced Intelligence Paradigms*, Vol. 10, Nos. 1/2, pp.146–159.

Biographical notes: B. Prabadevi is an Assistant Professor in the School of Information Technology and Engineering at VIT University, Vellore. She completed her BE in Computer Science and Engineering under Anna University, Chennai in 2010. She pursued her post-graduation in ME Software Engineering under Anna University, Chennai in 2012. She is currently pursuing her research at VIT University, Vellore in the area of network security attacks – ARP Cache Poisoning attacks. Her areas of interest are computer networks, network security, computer architecture and software engineering.

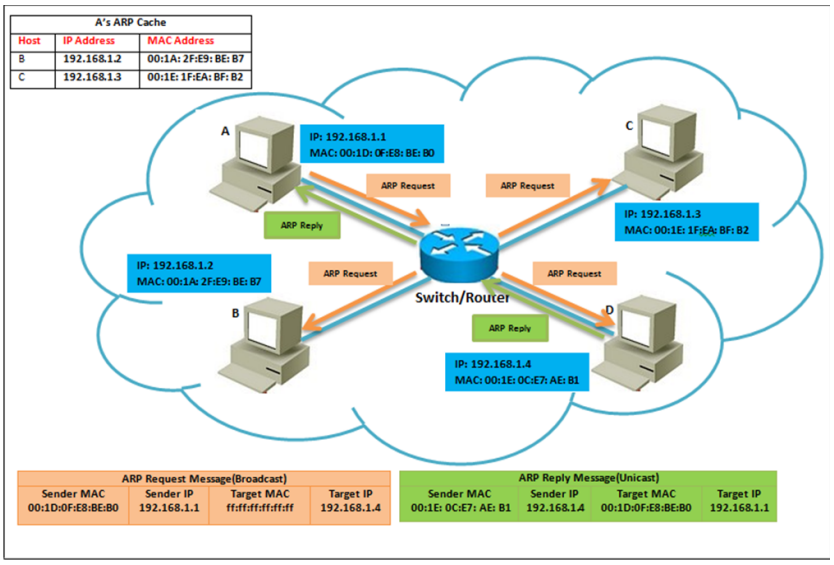
N. Jeyanthi is an Associate Professor in the School of Information Technology and Engineering, VIT University, India. She received her PhD and MTech in Information Technology with Networking as specialisation from VIT University, India in 2013 and 2006, respectively and BE in Computer Science and Engineering from Madurai Kamaraj University, Madurai, India in 1999. Her research interest is on network security in real-time applications. She has published around 25 international journal papers and many international conference papers. She received active researcher award from VIT University

for three consecutive years. She is an editorial board member of international journals and acted as program chair in many international conferences. She is a life member of Indian Society of Technical Education.

1 Introduction

ARP and RARP are the address resolution protocol and reverse address resolution protocol respectively. If two hosts wish to communicate then they must know the communicating entity's (IP, MAC) pair. This is not an issue if the communicating party knows either the network layer address (IP) or the link layer address (MAC), as the other can be obtained with the help of tele-communication protocols ARP or RARP. When A knows only the IP address of the communicating host B, the host A can obtain B's MAC address by broadcasting an ARP request query to all the other nodes in the network. Now it first checks the IP-MAC pair in the ARP cache. If found the communicating host will make use of the details in the entry otherwise a broadcast message is sent. On receiving the query if IP address match of B is found, B will send a unicast message in the form of ARP reply packet with its MAC address to A. When A knows only the MAC address of the communicating host B, the host A can obtain B's IP address in the similar way through RARP protocol. ARP and RARP protocols maintains a table of IP and MAC addresses of all hosts in the network, so if any request-reply made to these protocols is updated to all the hosts in the network. Each time a request is made by new host for IP-MAC address an entry is made in the ARP table of the host which maintains <IP, MAC> of hosts retrieved using ARP request and reply messages. The working principle of ARP is depicted in Figure 1. Since the ARP is used for servicing within a single network boundary, it is placed in the internet protocol suite's link layer.

Figure 1 ARP request reply procedure (see online version for colours)



ARP or RARP provides better service to its users if there are no compromised nodes in the network. The problem arises when a malicious host is on the network looking for IP/MAC address of victim to exploit it. The malicious node can devastate the entire network causing various ARP attacks through cache poisoning. The stateless nature of ARP protocol (Leon et al., 2015) accepts requests and responses irrespective of legitimacy checking for originality is the cause for the vulnerability. Some of the most prevalent ARP attacks include man-in-the-middle (MITM) attacks, MAC cloning or spoofing attacks, denial of service (DoS) attacks to the intended users (Prabadevi and Jeyanthi, 2014) and impersonating the legitimate host in the network (Ortega et al., 2009). Distributed DoS (DDoS) attacks occur when vulnerabilities to ARP affects the distributed systems connected in the LAN (Abad and Bonilla, 2007). DoS and its variants remain the root cause for most of the attacks to LAN through ARP information leakage. Normally MAC spoofing takes place when an attacker tries to impersonate a host which is been shut down by DoS attack.

The ARP cache entries expire based on the operating system used. Typically it expires after 20 minutes to billet the hosts with dynamic IP assignments (Abad and Bonilla, 2007). Whereas in some operating system the expiration timer is reset for every time an entry is made to the cache (Abad and Bonilla, 2007). When the request for MAC addresses arise from any host, first and foremost the corresponding host's ARP cache is checked for data, if match found the MAC address is automatically retrieved otherwise ARP request-reply process takes place.

Although various mechanisms for combatting these attacks in the form of hardware switches or software like intrusion prevention and detection systems, specialised tools, port security feature or enhanced ARP protocols are available, still there is no 100% solution to these attacks (Leon et al., 2015).

The document is organised as follows: Section 2 provides the brief introduction to ARP protocol, various attacks and mitigation techniques, Section 3 describes the proposed work, Section 4 provides the experiments results and screenshots and Section 5 concludes with further work.

2 Address resolution protocol

2.1 ARP – an overview

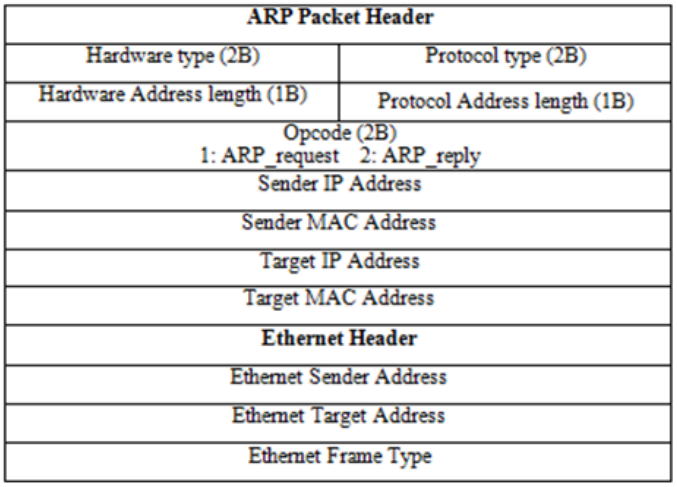
By design the telecommunication protocol ARP is stateless, because it will process any unicast reply though it has not sent corresponding broadcast messages over the network. This is the loophole for the attackers to sit on the network and exploit either a single host or even abolish entire network itself. The ARP packet format is specified in Figure 2. The ARP request and reply messages are:

1 when the host A wants to communicate with B it sends a broadcast ARP request message to B stating: "I am 'A.A.A.A' with 'aa:aa:aa:aa:aa:aa' (in sender's IP and MAC field) tell who is 'B.B.B.B' (destination IP)"

2 the host B with the IP 'B.B.B.B' replies A with a unicast ARP reply message in which the sender's field MAC is set to 'bb:bb:bb:bb:bb:bb'.

As per RFC 826 (Plummer, 1982), the IP-MAC pair received through unicast ARP reply will be mapped onto the host's ARP table only if the communicating host's IP address in its ARP table; otherwise the reply is thrown away. Some exceptional cases of ARP viz., gratuitous ARP and proxy ARP that works violating the procedure specified in RFC 826 (Abad and Bonilla, 2007).

Figure 2 Arp message on Ethernet/IP



In proxy ARP, the router will respond to the ARP queries on behalf of hosts on the other side irrespective of gateway configuration without the knowledge of the hosts. The gratuitous ARP is used by the hosts willing to join the network through dynamic IP address and to ensure that its dynamic IP is not been used by any other hosts in the network (Abad and Bonilla, 2007). Gratuitous ARP is the request message in which both sender and receiver IP address will be the same and destination MAC field will hold the broadcast address (Salim et al., 2012).

The protocol does not have any mechanism to retain the consistency between the addresses in ARP packet header and Ethernet header. The uncorrelated addresses in the headers remain unnoticed (Trabelsi, 2011). This remains more advantageous to the attackers.

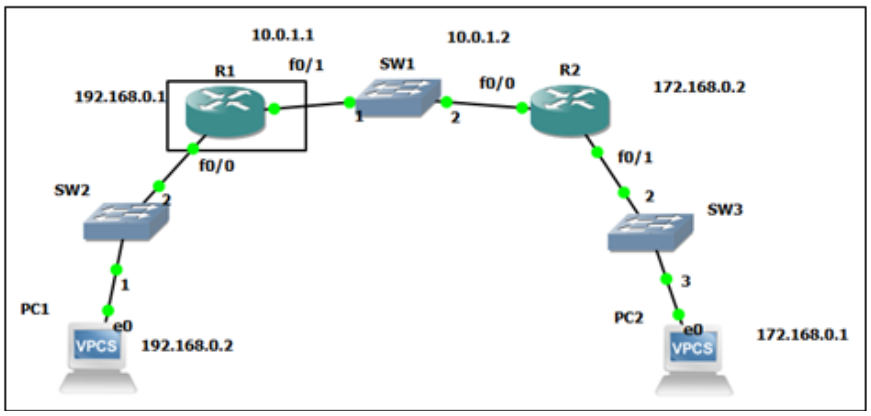
There are two types of entries for ARP table or cache viz., static and dynamic. The entries in the former remain unchanged until next system boot, whereas in the latter the entries keep changing frequently.

2.2 Experimental analysis of traditional ARP

The working procedure of ARP is analysed by using network simulator GNS3 1.4 Orc3 and packet analyser Wireshark 1.12.7. The initial network configuration is as follows: two VPCS with IP addresses 192.168.0.2 and 172.168.0.1 connected to R1 (192.168.0.1) and R2 (172.168.0.2) respectively. The VPCS are connected to routers via Ethernet switches (Ethernet0) which are dynamic emulated Ethernet switches. The gateway

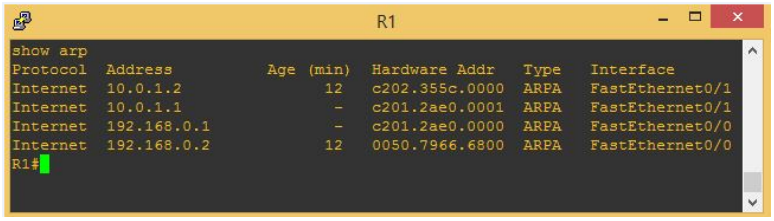
addresses are 10.0.1.1 and 10.0.1.2. R1 and R2 are CISCO routers c3725. The switches and routers are connected using fast Ethernet ports f0/0 and f0/1. The network setup is depicted in Figure 3.

Figure 3 Initial network setup in GNS3 (see online version for colours)

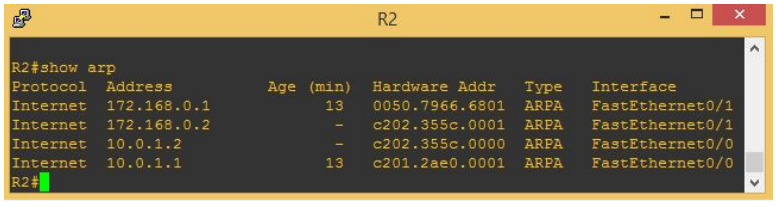


All the devices are connected and the routers are configured. Static routing is performed, so that whenever PC1 and PC2 want to communicate, it can be done via R1 and R2 respectively. Now Wireshark is used to capture all the packets following from PC1 to destination. Before starting the Wireshark the ARP caches of R1, R2, PC1 and PC2 are cleared. The contents of the ARP cache of R1 and R2 are shown in Figures 4(a) and 4(b) respectively.

Figure 4 (a) ARP cache R1 (b) ARP cache R2 (see online version for colours)



(a)

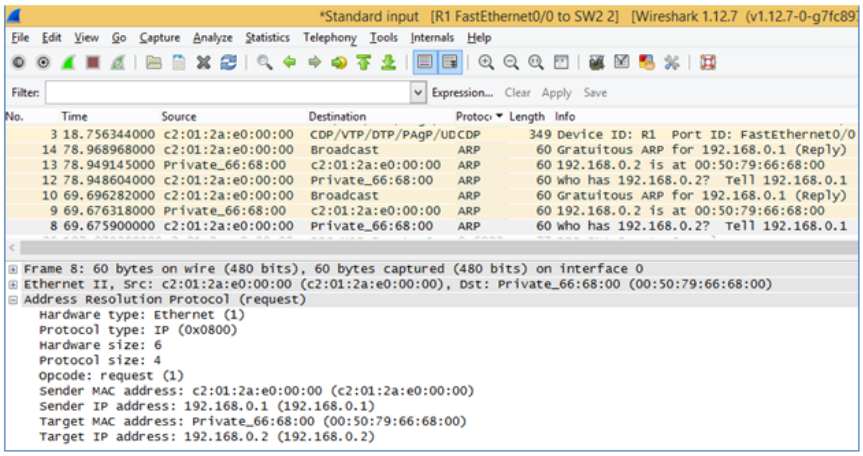


(b)

Notes: Figure 4(a) depicts the ARP cache contents of router R1. This gives the information about the hardware addresses and interfaces of the nodes.
Figure 4(b) depicts the ARP cache contents of router R2. This gives the information about the hardware addresses and interfaces of the nodes.

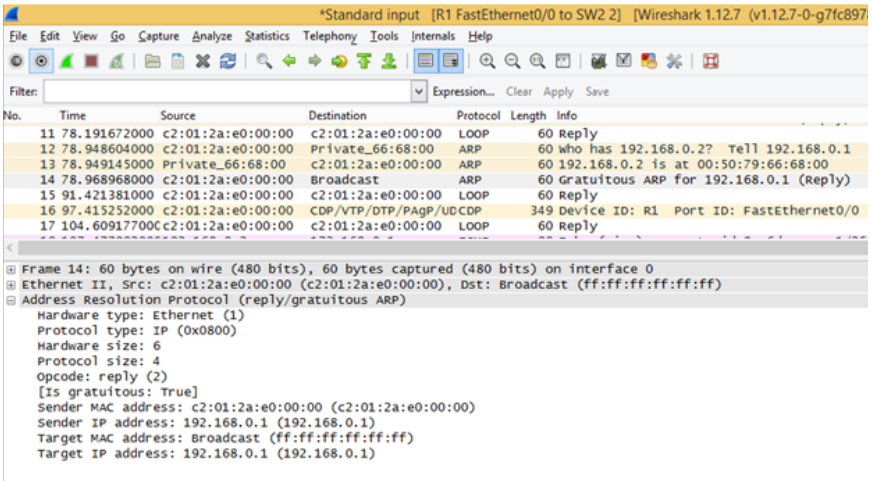
The ARP request and reply messages transferred among various nodes in the network is depicted in Figure 5 and Figure 6. This is analysed when PC1 pings to PC2. The communication takes place via routers. The figures Figure 5 through Figure 7 depicts how the host in a network introduces itself through gratuitous ARP packets and after updating the ARP table, the ICMP ping messages are transferred from PC1 to PC2 as in Figure 8.

Figure 5 ARP request message from PC1 to router R1 (see online version for colours)



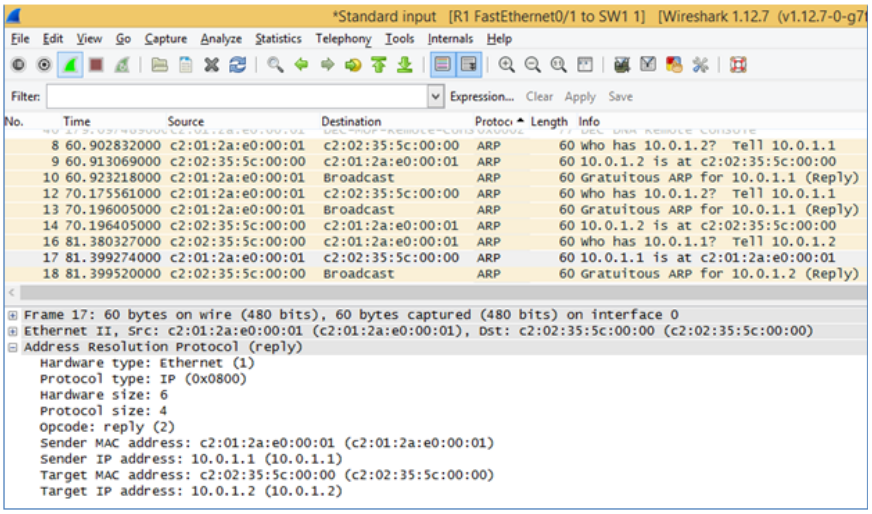
Notes: Figure 5 depicts the ARP request from the Router R1 (192.168.0.1) to PC1 (192.168.0.2). The bottom of the figure details the ARP request message.

Figure 6 Gratuitous ARP reply message to broadcast host information (see online version for colours)



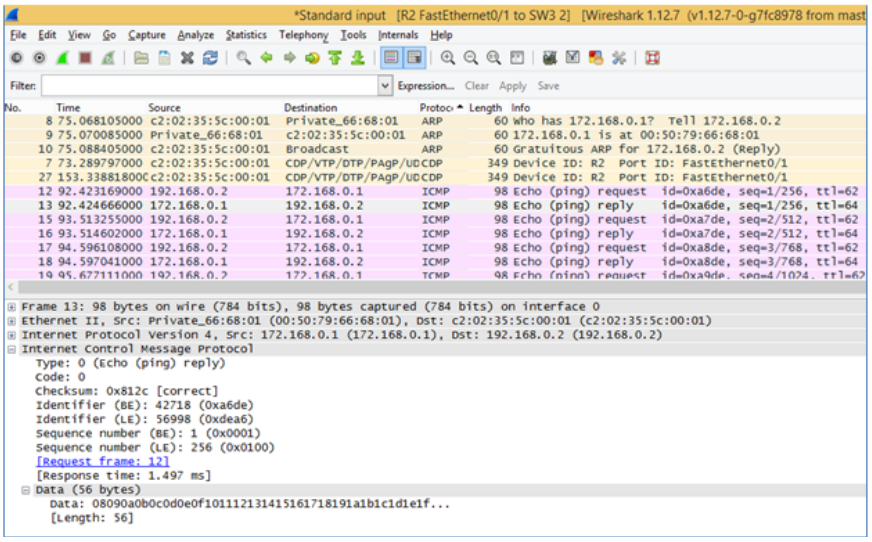
Notes: Figure 6 depicts the fratuitous ARP reply by R1. This is a broadcast message of R1 192.168.0.1, which will be delivered to all the nodes in the network and the nodes will update their ARP table with this information.

Figure 7 ARP reply message with MAC information (see online version for colours)



Notes: Figure 7 depicts the ARP Reply message from Gateway 1 (10.0.1.1) to Gateway 2 (10.0.1.2). Now two networks are well-known to each other. The two PCs PC1 and PC2 can ping now, i.e. they can pass the messages from one to another.

Figure 8 ARP messages from PC2 and ICMP ping messages (see online version for colours)



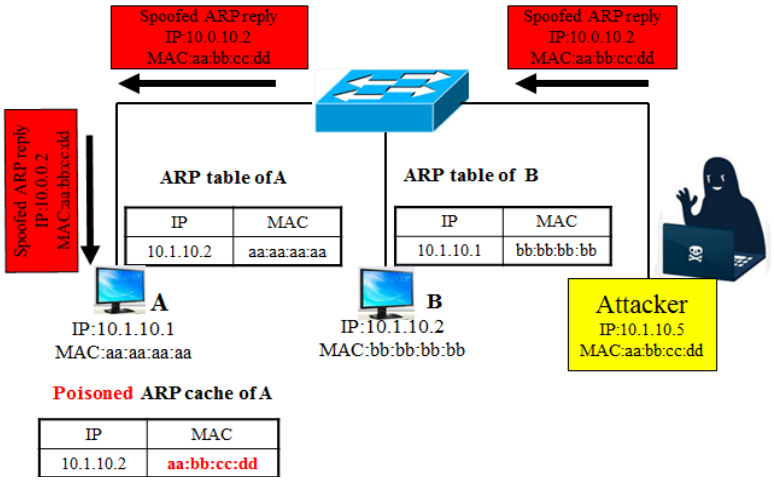
Notes: Figure 8 depicts the messages transferred between PC1 and PC2. This is PC2's Ping reply to request sent by PC1.

The ARP poisoning to the routers can be done and analysed by using Ettercap with GNS3 and Wireshark.

2.3 ARP cache poisoning or ARP spoofing attacks

ARP cache poisoning refers to poisoning the ARP table when a malevolent host makes an incorrect IP-MAC pair entry onto its table. It is also termed as spoofing when then poisoned or spoofed entry is used exploit the other host's ARP cache.

Figure 9 ARP cache poisoning attack (see online version for colours)



The ARP cache poisoning attack is depicted in Figure 9. The attacker first makes a forged entry with spoofed address; the target host considering it as a legitimate response updates its own cache though the correct one already exists. The malevolent host can even send some ARP reply with spoofed IP-MAC bindings to corrupt the victim's cache. This happens when OS accepts the request without guarantying that a prior request was sent for this reply (Trabelsi, 2011). Because of this feature of ARP it is vulnerable to devastating attacks.

2.3.1 ARP poisoning-based MITM and DoS attacks

MITM attack is usually performed to snort the traffic in the network; consequently exploits the victims' system by redirecting to malevolent host. This is executed as follows: the attacker uses some sniffers to capture the packets and redirects them to the destined receiver. The actual source and destination host is not aware of this activity. MITM is depicted in Figure 10. The ARP table entry with Red colored text indicates the forged MAC addresses. It takes place as follows:

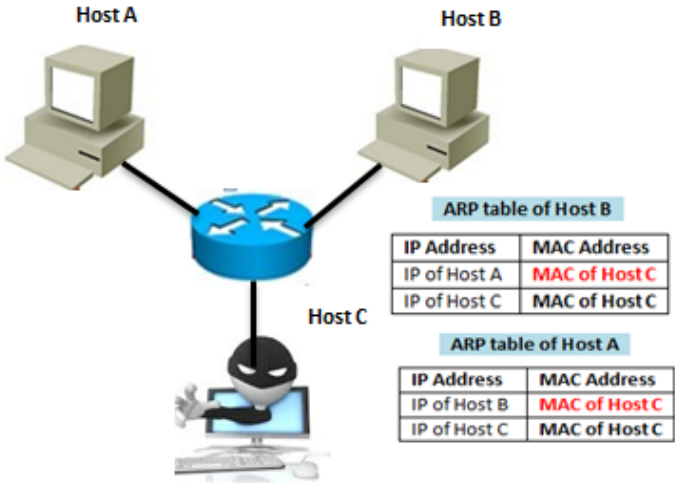
1 The attacker C first sends an ARP request message with B's IP and C's MAC.

2 Now A's cache is updated with bogus entry as (B's IP-C's MAC) pair replacing the actual IP-MAC pair of B.

3 When a sends unicast reply to B, it will be destined to C instead of B because of bogus entry.

4 C's cache is updated with A and B's information. It can now monitor the communications between A and B.

Figure 10 ARP-based MITM attack (see online version for colours)



In this scenario, the attacker can poison the ARP cache of victims by sending the fake packet, without redirecting. In turn the victims will be sending their messages to unintended destination. DoS can be mounted, since A's cache is exploited by C and A's message is not received by the intended receiver B. If the attacker plays the same role by sending bogus ARP request packets from more than one compromised nodes, the victim 'A' can deny its services to all the other intended hosts in the network. This is the variant of DoS, DDoS attacks.

2.3.2 ARP poisoning-based host impersonation

By ARP spoofing, the attacker can impersonate as host by responding to other hosts trying to communicate with the victim host. By this the victim host may not be providing legitimate services to its intended users.

Apart from the attacks mentioned there are some more attacks like broadcast attacks, MAC spoofing attacks caused by ARP Spoofing. The attacker can spoof the MAC address of the victim which is already been under DoS. Here the attacker hiding his own identity by MAC spoofing/cloning (Salim et al., 2012) can receive or even send messages to other nodes in the network. The attacker can advertise fake services.

Several solutions to detect and prevent this ARP poisoning attacks exists such as monitoring the ARP cache, ICMP ping restriction (Kumar and Tapaswi, 2012; Chomsiri, 2008), enhancements to ARP protocol like S-ARP (Bruschi et al., 2003), E-ARP (Nam et al., 2010), TARP (Lootah et al., 2007), GDPS (Salim et al., 2012) using gratuitous ARP packet, S-DHCP and anti-dote approach (Abad and Bonilla, 2007). The solution to overcome ARP poisoning-based MITM, a centralised ARP server is used and it successfully prevents MITM (Kumar et al., 2012). E-ARP prevents gateway MAC poisoning attack by conflict resolution using voting mechanism but consumes more memory.

3 Proposed method

This section provides a method for detecting ARP cache poisoning attacks viz. MITM attacks, DoS attacks and host impersonation. The ARP spoofing can be done by four ways (Yang et al., 2014) as follows:

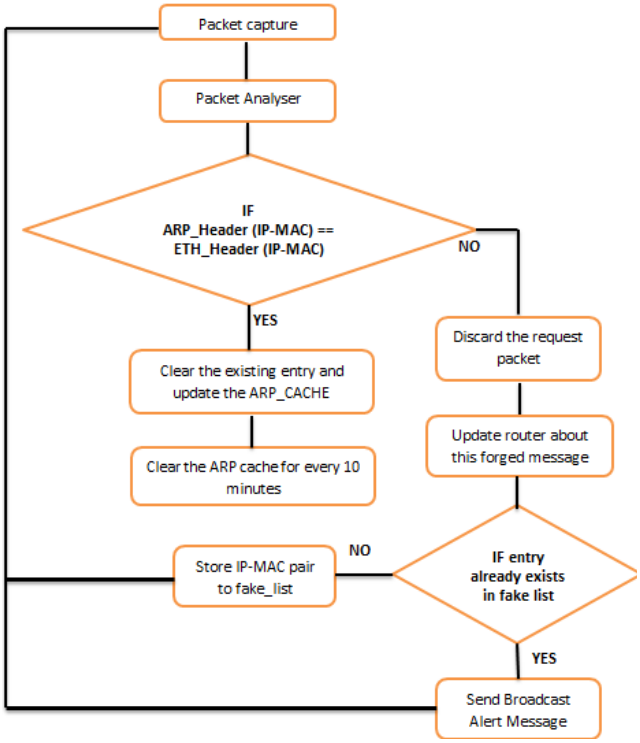
1 Counterfeiting internetworking device gateway to spoof other computers on the network by fake ARP request and ARP reply packets to vanish entire network. This may even lead to DoS attack.

2 The attacker impersonates the other host, by sending spurious packets to the gateway and makes fake entry in the gateway's ARP cache.

3 Using one host to fake another computer and launch an MITM attack.

4 Flood attacks/broadcast attacks.

Figure 11 Flow chart for proposed method (see online version for colours)



The proposed framework in Figure 11 has the following:

1 packet capture and analysis

2 Update_ARP cache

3 Update_Fake_list

4 Broadcast_ARP_Alert_Message.

3.1 Packet capture and analysis

Consider Figure 1 where the host A would like to share its data with host D, but it has D's IP address only. In this case A makes use of ARP Request message to get D's MAC address. It broadcasts ARP Request message with this details in the packet:

Figure 12 ARP broadcast request message (see online version for colours)

Sender-IP : IP-A	**Sender-MAC : MAC-A**
Target-IP : IP-D	**Target-MAC: ff:ff:ff:ff:ff:ff**
ICMP Message	**"Who owns this IP-D?"**
Opcode= 1	

Here the target MAC is specified with broadcast address ff:ff:ff:ff:ff:ff. On receiving this packet, this module captures the packet and first ensures it is an ARP packet by applying filters.

Working Procedure:

Variables: arp_IPA,Eth_IPD,Eth_IPD. arp_IPD, arp_MACA, arp_MACD, Eth_IPA, Eth_IPD

When A wants to know D's MAC:

// sender → arp_IPA,arp_MACA and Eth_IPA, Eth_MACA

// target → arp_IPD, arp_MACD and Eth_IPD, Eth_MACD

Step 1:

send ARP_Request_Message (ARP Header, Ethernet Header); //opcode = 1

Step 2: /*At D*/

packet_cap_analysis();

Decode_the _packet();

Step 3:

if ((arp_IPA==Eth_IPA) && (arp_MACA==Eth_MACA))

Update_arp_cache();

send_ARP_Reply_Message (ARP Header, Ethernet Header); //opcode=2

else

Store the forged IP-MAC_address into fake_list;

send alert message to the_Router or gateway;

if(arp_IPA and arp_MACA exists in fake_list)

send Broadcast_Alert_Message;

Step 4 : /*At A*/

// sender→ arp_IPD, arp_MACD and Eth_IPD, Eth_MACD

//target→ arp_IPA,arp_MACA and Eth_IPA, Eth_MACA

Repeat Step 2 and Step 3;

Step 5: Clear the ARP-Cache for every 10 minutes

The crux of this module is, it checks whether the IP-MAC pair received in the ARP header and Ethernet header matches. Based on the match result, it will make a call either to Update_ARP_cache module or Update_Fake_list module and sends a message about this mimicking to the router. So the router either redirects the packet or drops it.

3.2 Update_ARP_cache and Update_Fake_list

If the ARP header IP-MAC pair and Ethernet header IP-MAC pair matches, the host D updates its ARP table and generates a unicast reply message as:

Figure 13 ARP unicast reply message to A (see online version for colours)

Sender-IP : IP-D	**Sender-MAC : MAC-D**
Target-IP : IP-A	**Target-MAC: MAC-A**
Opcode= 2	

If the entry does not match, then it is added to fake_list. This fake list is updated every time a forged message arrives. If a fake message arrives more than once then Broadcast_Alert message is called to deliver this information to every node in the network.

This feature helps to avoid the future attacks on other victims in the same network by the same group of handlers. The address will be updated to cache if it is by Gratuitous ARP packet. This ensures authenticity to ARP messages.

3.3 Broadcast_Alert_Message

If the ARP header IP-MAC pair and Ethernet header IP-MAC pair does not match, then D will discard/ drop the packet and generate a unicast message to its router about the fake pair. Meanwhile it maintains a list of fake IP-MAC pair. For each time it checks the fake-list for IP-MAC pair, if it matches then generates a Broadcast_Alert_Message to be sent over the network to avoid other hosts from updating its ARP cache with this fake packets. A sends this message continuously with certain frequency, so that it can reach the hosts in the network before the attacker tries to launch any attacks.

By this the MITM and host impersonation can be avoided which in turn avoids the DoS attacks. This will be safe because the router or gateway will be intimated about the fraudulent source by a unicast message, which will also be cross checked for legitimacy.

When a new node is linked to the network, gratuitous ARP packet is sent to update the hosts in the network with new host's information. To avoid attacks through gratuitous ARP, these packets are also captured and analysed before updating the other hosts and gateways ARP cache.

In traditional ARP, the cache is cleared for every 20 minutes irrespective of the operating system, this can be reduced to 10 minutes and after each check for new ARP request, the existing entry can be replaced with new one only if it satisfies the constraints. This time limit can be further increased or decreased based on the nature of the network i.e. either static or dynamic.

In the proposed method the Ethernet header and ARP header are compared for assuring the authenticate reception of packets. In addition it maintains a fake list which is needed for sustaining the long term security in the network until it is destroyed. This fake list entry is cleared only when the network communication vanishes or for the long lived connections it will be automatically cleared based on the communication. This is to avoid the extra cost incurred in maintaining the fake list.

4 Conclusions and future work

This paper provides a brief overview of ARP poisoning attacks with demonstration of traditional ARP using GNS3 and Wireshark. The ARP poisoning attacks can be examined using Ettercap or Wincap. Based on the working principle of ARP and attacks to its environment, a simple procedure to prevent these attacks has been proposed. The proposed work will be cost effective and avoids host impersonation, DoS attacks and MITM attacks caused by ARP cache poisoning. Since it clears the cache every 10 minutes, the ARP table has to be updated each time a new host communicates in the network. Though this can be a loophole for the attackers, the checking of Ethernet header information with ARP header information, they will go vain without any information being retrieved. This will be an authenticated method which avoids most of the attacks by cache poisoning.

The future work is to implement the procedure proposed model by enhancing the packet format in the ARP protocol and to evaluate this on a live network.

References

Abad, C.L. and Bonilla, R.I. (2007) 'An analysis on the schemes for detecting and preventing ARP cache poisoning attacks', *27th International Conference on Distributed Computing Systems Workshops by IEEE Computer Society*, p.60.

Bruschi, D., Ornaghi, A. and Rosti, E. (2003) 'S-ARP: a secure address resolution protocol', *19th Annual. IEEE Conference Computer Security Applications*, pp.66–74.

Chomsiri, T. (2008) 'Sniffing packets on LAN without ARP spoofing', *Third International Conference on Convergence and Hybrid Information Technology, ICCIT'08*, IEEE, Vol. 2, pp.472–477.

Kumar, S. and Tapaswi, S. (2012) 'A centralized detection and prevention technique against ARP poisoning', *2012 IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 26–28 June 2012, pp.259–264.

Leon, E., Reyes Daza, B.S. and Octavio, J.S.P. (2015) 'Comparison between safety and efficient security of the ARP protocol', *8th ACM International Conference on Security of Information and Networks*, pp.318–321.

Lootah, W., Enck, W. and McDaniel, P. (2007) 'TARP: ticket-based address resolution protocol', *Elsevier*, Vol. 51, No. 15, pp.4322–4337.

Nam, S., Kim, D. and Kim, J. (2010) 'Enhanced ARP: preventing ARP poisoning based man-in-the-middle attacks', *IEEE Communications Letters*, Vol. 14, No. 2, pp.187–189.

Ortega, A.P., Marcos, X.E., Chiang, L.D. and Abad, C.L. (2009) 'Preventing ARP cache poisoning attacks: a proof of concept using OpenWrt', *IEEE Symposium on Network Operations and Management, LANOMS*, pp.1–9.

Plummer, D. (1982) *An Ethernet Address Resolution Protocol*, RFC 826.

Prabadevi, B. and Jeyanthi, N. (2014) 'Distributed denial of service attacks and its effects on cloud environment – a survey', *The 2014 IEEE International Symposium on Networks, Computers and Communications (ISNCC*), pp.1–6.

Salim, H., Li, Z., Tu, H. and Guo, Z. (2012) 'Preventing ARP spoofing attacks through gratuitous decision packet', *2012 11th IEEE International Symposium on in Distributed Computing and Applications to Business, Engineering & Science (DCABES)*, pp.295–300.

Trabelsi, Z. (2011) 'Hands-on lab exercises implementation of DoS and MiM attacks using ARP cache poisoning', *ACM International conference on Information Security Curriculum Development, InfoSecCD '11*, Kennesaw, Georgia, pp.74–83.

Yang, M., Wang, Y. and Ding, H. (2014) 'Design of WinPcap based ARP spoofing defense system', *2014 Fourth IEEE International Conference on Instrumentation and Measurement, Computer, Communication and Control*, pp.221–225.