International Journal of Engineering Sciences Paradigms and Researches (IJESPR)
(Vol. 35, Issue 01) and (Publishing Month: October 2016)
(An Indexed, Referred and Impact Factor Journal)
ISSN: 2319-6564
www.ijesonline.com

Detecting and Preventing Network Address Spoofing

Hamza A. Olwan[1], Mohammed A. Babiker[2] and Mohammed E. Hago[3]
University of Khartoum, Sudan
olwan777@gmail.com[1], moh_teg821@hotmail.com[2] and melzain88@gmail.com[3]

Publishing Date: October 12, 2016

Abstract
Malicious users can exploit the design weaknesses of the internet. The IP spoofing are on rise, the attacker's sole purpose is control of the particular device and then data acquisition and modify to his favor,This is achieved by exploiting the vulnerabilities in network protocols addresses, the ARP spoofing is the most popular attack in network,for this, it becomes very important against this type of attack by setting up system able to detect it , among the existing IDs ,we find the SNORT which is the most used, and we propose a method to detect and prevent ARP spoofing by expanding the ARP cache table to include the public keys.
Keywords: *Spoofing attack, ARP Spoofing attack, snort.*

I. Introduction

Network address spoofing is the most popular attack that can affect the performance of networks and it use to monitor, stop and redirect the traffic between two or more hosts that communicates with each others. It can be divided in two types:

- IP spoofing: in IP spoofing the attacker can gains unauthorized access to computer of network by making it appear that a malicious message has come from a trusted-machine by spoofing the IP address of the machine.
- ARP spoofing: in this type the attacker interrupt the request when the machine send an ARP request to get the layer 2 (MAC) address by predefined layer 3 (IP) address, and then the attacker answer with their MAC address, which the other machine use it to send a packages, so any traffic send it receive by an attacker and doesn't arrive to desired destination.

ARP is built in trusting all the nodes in network. It is high-efficient, but not very safe. There are existing two main shortcomings. ARP is stateless which mean that ARP reply packages can be send free and ARP reply do not need to be certified. This unconditional trust mechanism make it possible for ARP spoofing, on the other way ARP cache table is update dynamically, there is out-of-services time. The ARP attack can be carried out, if the malicious users change the cache table in victim host before the next change.

Type of ARP Spoofing Attacks

1. **Denial Of service(Dos) attack**

In this way communicating originating from host is blocked by the attacker. When ARP spoofing is used to change the ARP cache table of host, then every packet send by host is directed to the attacker.

2. **Host impersonation attack**

Here the idea of this type after receiving the packet from host attacker responds to it and creates impression that host is communicating with desired destination.

3. **Man-In-The-Middle (MITM) attack**

This type applied by poisoning ARP cache of two hosts which are communicating with each, the attacker can monitor all the traffic between two hosts. The attack is used to access sensitive information like passwords and modify the data being compromising the data integrity.

II. Snort

Snort was created by Martin Roesch in 1998 .As most open –source projects, it started out as a small –scale application made just for fun as an alternative to the full blown commercial intrusion detection system. It is open source network intrusion detection and prevention systems it can be analyze real-time traffic analysis and data flow in network. Also it is able to check protocol analysis and therefore it can detect different type of attack. [1]

Operation modes of snort

- Sniffer mode
- Packet Logger mode
- Network Intrusion Detection Mode

In our implementation we use Network Intrusion Detection mode.

III. Related Works

There are so many researches that put solutions for networks address spoofing (IP-ARP) attacks.

Yunji Ma [2] propose an effective methods to preventing IP spoofing attacks based on trusted network and cooperation with trusted adjacent node. When a machine (M1) send a request to machine (M2) the machine (M2) send a traceroute request to adjacent node (C), the node (C) trace the information route and send the result to M1. If the M2 is reachable then the request accepted otherwise the request is rejected.

Jesus, Mohd and James [3] in their propose method the end-hosts send their packets to their corresponding Access Router. The access routers are required to send a copy of each packet they receive to a judge router, JR, and the original packets to their corresponding distribution router. Also, the ingress/egress, Rie, router is required to send a copy of the packets it

Receives from hosts within its AS to JR. JR starts a timer and counts the number of packets that it receives at each interface until the timer expires. Then JR marks a packet as valid if its source address belongs to the range of IP addresses assigned to the forwarding access router's sub network; if not, JR marks the packet as spoofed. Trust is calculated based on the number of spoofed packets.

NIDS monitor the traffic of entire network. Dr. S. G. Bhirud and Vijay [4] in their propose method a small agent program called information agent (IA) installed in every machine of the organization. NIDS communicates with these agents to detect spoofing activities.

IP address and identification is used to uniquely identify every package sent by a host. So this field is used in this mechanism to detect spoof message send by zombie machine. NIDPS uses IP-MAC address pairs sent by lAs to create and manage table of IP-MAC address pairs of all machines in LAN.

Alberto and Marcelo [5] propose extending FCFS SAVI to prevent MAC duplication as follows: FCFS SAVI Bridge is modified to include IP/MAC pairs associated with physical port. Only packages come from physical port with the existing binding of IP and MAC address is forwarded.

Opeyemi Osanaiye [6] suggest using the OS fingerprint on an IP package, the advantage of that different OS have their unique value combination for TCP\IP header field is exploited. The incoming IP header is analyzed and compares the result with the P0F database.

Their suggest approach has two phases:

Passive: compare the header info With the P0F.

Active: send package to source IP to get information about the source host and compare it with the previous info in P0F database.

IV. Compare between the existing methods

Methods	advantage	disadvantage
Defense IP spoofing using trust adjacent node	Detect IP spoofing attack when using unreachable IP	Cannot detect attack when the IP is reachable
Detect IP spoofing using Judge router (JR)	Detect IP spoofing is more efficient in this approach.	Require a more devices. Every AS (interconnect host) need additional device called judge router (JR). More burden to AS and Rie router.
Using information agent (IA) with NIDS to detect IP-ARP spoofing	Detect ARP-IP attack and ARP cache table poisoning.	NIDS must be more powerful and high performance to overlap with

	- Easy to implement	repeated IA check
Extending FCFS SAVI.	Difficult to spoofing IP-MAC address remotely	Prone to physical attack when the attackers change their MAC address
Using OS fingerprint on IP packages	-Easy to implement. -No more additional devices need.	The fingerprint may equal to other device use the same edition and copy of operation system

V. Detection of the ARP-Spoofing Attack by Snort

We propose a simulation of the *ArpSpoofing* attack to see how attack is occurring and how SNORT reacts:

1. Presentation of the topology

To implement our proposed model we firstly construct the topology of the network which is consisting of four machines: Intruder machine, M1 machine, M2 machine and Sensor machine. The machines (M1) and (M2) communicate directly, the intruder wants to poison the cache of the machine (M1) which is dubbed victim machine, to steal the identity of machine (M2) known as target machine. On the other hand the Sensor machine monitors the inbound and outgoing traffic and therefore makes alert when it found intrusion.

2. Parameter setting of SNORT

There are many parameters that must be set when detecting Arp- Spoofing. Here We setup snort NIDS in the sensor machine to detect the Arp-Spoofing attack and we have to put the IP address and the MAC address of the machine target in **snort.conf** file with the directive arpspoof_detect_host
Preprocessor arpspoof [: unicast]
Preprocessor arpspoof_detect_host:192.168.174.146 at 00:0C:29:78:a6:e6

This pair of address IP/MAC is placed in a table which will be used to detect the Arp -spoofing attack. If there are several target machines, it is necessary to declare several lines of Preprocessor arpspoof_detect_host with their IP and MAC addresses. Then, SNORT had to run in NIDS mode with the directory of the configuration file to use:
#snort -c /etc/snort/snort.conf

3. Execution of the attack

To execute an Arp-Spoofing attack, we install dsniff in intruder machine and use the tool "arpspoof", which is executed in the Intruder machine:
arpspoof –t victim_address target_address
#arpspoof –t 192.168.174.145 192.168.174.146

The packets sent are ARP packets Replay. This packets are poisoning the cache of the machine (M1) 192.168.174.145 indicating that MAC address associated with 192.168.174.146 is now 00:0C: 29: 1b: e8: dc.

4. Detection of the attack and SNORT's reaction

The Sensor machine detects the attack with the preprocessor. If the sender hardware address isn't the same addressee in the configuration file then an alarm message is generated. SNORT logs the alarm message in the /var/log/snort/alert file.

SNORT generates a large quantity of reports that requires time and effort from the network administrator, to decide which a true intrusion is and which is a false positive. The reaction of the administrator is usually late, in this time the intruder could cause enough damage and even leave the system, and in this case the intrusion detection in real time isn't useful because it has not prevented the intruder to achieve his goal. [7]

VI. Prevention ARP spoofing enhancement

In our suggestion enhancment we expand the ARP cache table in any host to include the public key in addtion to IP and MAC in the table.
Assumption: host (A) want to comunicates with host (C), the phases are :

A. host (A) send ARP request on network see figure (1).
B. The desired host (C) recive ARP request and do some operations :
1. Encrypt the MAC_c using their private key $E(PR_c,MAC_c)$.
2. Add the encrypted MAC to ARP header.
3. Send the ARP response to host (A).

C. Host (A) recive ARP response from host (C), see figure (2) and do some oerations:
1. Acquire the public key of the host(C) from ARP cache table using IP address.
2. Decrypt encrypted MAC address using the PU_c $DE(PU_c, MAC_c)$
D. Compare the result of decryption with MAC address in ARP cache table according to IP address.
E. If the decrypted MAC address equal to MAC address in ARP cache table accept the connection and exchange data.
F. Otherwise , terminate the connnecction and send alert to NIDS in network or to users.

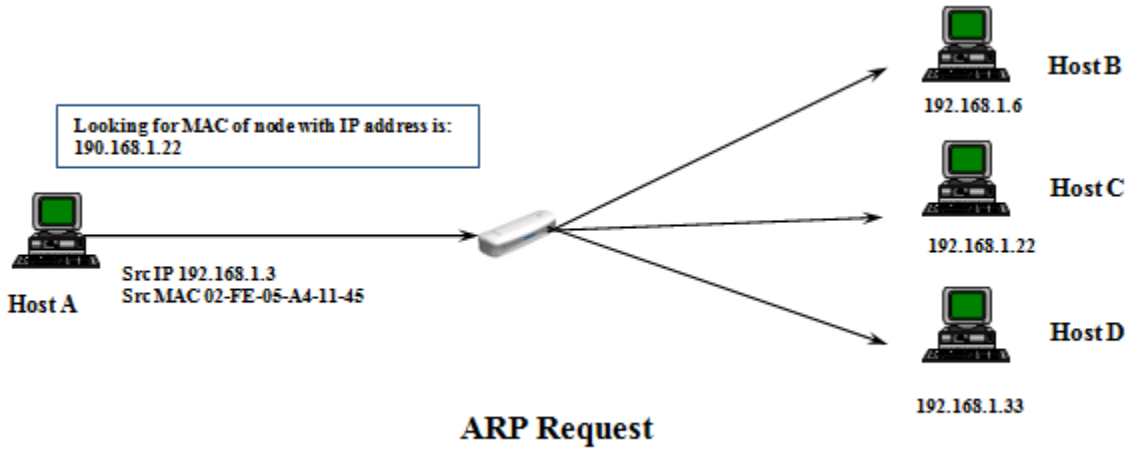

Figure 1: Host A send ARP request to network

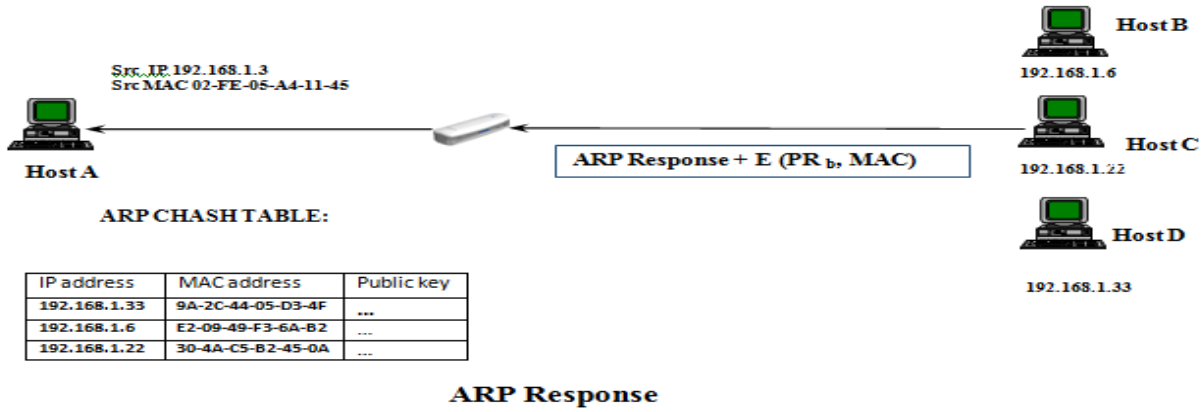

Figure 2: The desired host C sends ARP response

VII. Operation of Enhancement ARP Spoofing Flowchart

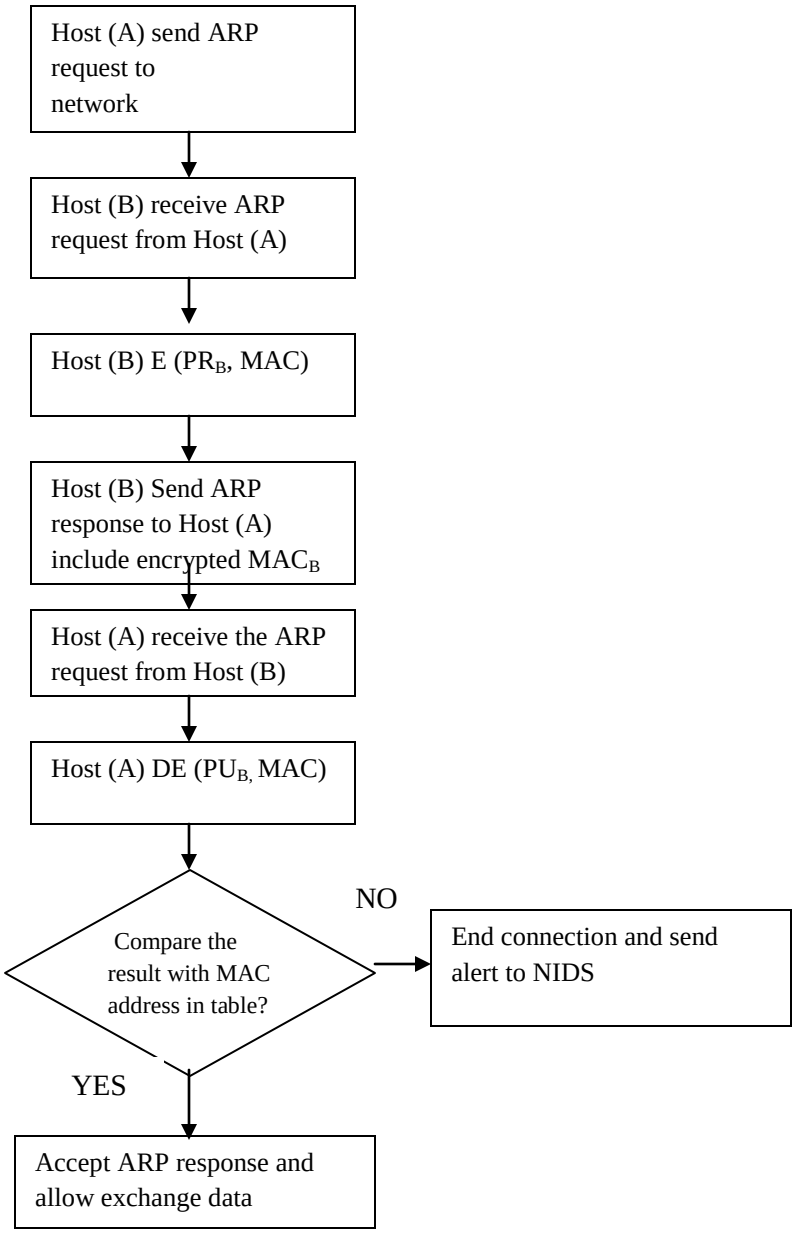

Figure 3: Flowchart of ARP spoofing detection and prevention

VIII. Conclusion

In this paper we reviewed different methods used to detection and prevent the IP-ARP spoofing attack during DDoS attack and poisoning ARP cache table and others network address spoofing, we propose a SNORT NIDS to detect and prevent ARP spoofing and enhance of ARP spoofing detection and prevention by using public encryption schema with expanding the ARP cache table to include the public key. In future work, the plan is to implement the suggested ARP spoofing detection enhancement.

References

[1] Kumar, V., & Sangwan, D. O. P. (2012). Signature Based Intrusion Detection System Using SNORT. International Journal of

Computer Applications & Information Technology.

[2] Ma, Y. (2010, September). An effective method for defense against ip spoofing attack. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on (pp. 1-4) IEEE.

[3] Gonzalez, J. M., Anwar, M., & Joshi, J. B. (2011, July). A trust-based approach against IP-spoofing attacks. In Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on (pp. 63-70). IEEE.

[4] Bhirud, S. G., & Katkar, V. (2011, November). Light weight approach for IP-ARP spoofing detection and prevention. In Internet (AH-ICI), 2011 Second Asian Himalayas International Conference on (pp. 1-5). IEEE.

[5] García-Martínez, A., & Bagnulo, M. (2012). An Integrated Approach to Prevent Address Spoofing in IPv6 Links.

[6] Osanaiye, O. (2015, February). Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing. In Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on (pp. 139-141) IEEE.

[7] Boughrara, A., & Mammar, S. (2012, March). Implementation of a SNORT's output Plug-In in reaction to ARP Spoofing's attack. In Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on (pp. 643-647) IEEE.