ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Study of Vulnerabilities of ARP Spoofing and its detection using SNORT

Rajneet Kaur Bijral
Department of Computer Science & IT
University of Jammu, J & K
Jammu, India

Alka Gupta
Department of Computer Science & IT
University of Jammu, J & K
Jammu, India

Lalit Sen Sharma
Department of Computer Science & IT
University of Jammu, J & K
Jammu, India

Abstract **-**As the organizational dependency on information exchange has increased exponentially the threats to information security have also increased. The organization has to spend a generous amount of resources to protect the information from the intruders. Attackers exploit the vulnerabilities of the network applications to have unauthorized access to organizational information. These vulnerabilities come into being because of poor application code or design. In this paper we will be discussing one such type of design level vulnerability that is *ARP Spoofing*. *ARP Spoofing* has been used to perform many types of attacks, including session hijacking, and cookie hijacking. We have used Snort for detecting *ARP Spoofing*. Performance of the snort for detecting *ARP Spoofing* has been studied experimentally on a real network by varying number of the target.

Keywords*:* Address Resolution Protocol, ARP Spoofing, MAC Address, Man in Middle Attack, SNORT

I. INTRODUCTION

The ARP (Address Resolution Protocol) is the protocol which is used to map logical address with the physical address. Whenever an incoming packet designated for a particular host arrives at the gateway, the gateway requests the ARP program to find the physical address of that host. If the *ARP cache* does not find the host entry in the *ARP table* then ARP broadcasts the *ARP request* to all the host in the network in order to find the MAC address associated with the requested host. The machine which recognizes the broadcast IP address as its own sends the *ARP reply*. *ARP table* is then updated and incoming frame is sent to the corresponding MAC address. The stateless property of ARP is exploited by attacker to perform *ARP spoofing*.

Snort is an open source intrusion detection and prevention system, capable of analyzing real time traffic and logging packets on IP networks. It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes[1]. It sniffs the packets in the network and compares all incoming packets with its rule-set to identify any malicious activity. Snort can also be used to detect *ARP spoofing* and identify the MAC addresses of the malicious hosts in the internal network which are sending the spoofed packets. In this paper, we have performed *ARP spoofing* by exploiting the vulnerability of ARP. Attacks like cookie stealing and session hijacking have been performed using *ARP spoofing* in the Local Area Network. Snort has been used to detect *ARP spoofing* in the internal network and Snort's performance has been tested by varying the number of target hosts from one to five.

II. RELATED WORK

Several solutions have been proposed for solving the problem of *ARP Spoofing*, some of them are enlisted:

S-ARP model is based on encryption, in which every host has own public/private key which they send to Authority Key Distribution(AKD) , with the help of which encryption and decryption take place. One of the major disadvantages of this method is that each time it encrypts the message hence it takes time for every transmission as a result of which performance decreases [2]

Kernel-based patch: Anticap [6] and Antidote [7] are solutions to ARP spoof they suggested a patch to some specific OS to provide protection against ARP spoof. But these patching can be used with a specific kernel. Moreover, it slows down the performance of the overall network.

Xing et al. [8] used WinPcaP library to capture andfilter ARP packets. ARP response packet are received and compares against the correct IP–MAC address pairs and updated if it is correct.

Ai-zeng Qian[9] proposed static entries to prevent *ARP spoofing* but the technique still doesn't work with dynamic networks. In this method, the administrator assigns all IP addresses along with their MAC to the server so it will be not feasible for large scale network.

Vinay K.R *et.al* [10] This paper discusses *ARP spoofing* attack. The paper proposed an efficient algorithm based on ICMP protocol to detect *ARP spoofing* attack. In this

technique the ARP packets are collected and analyzed, and then ICMP echo request packets are used to probe malicious host according to its response packets.

Sumit Kumar *et.al* [11] This paper discuss a centralized system and ARP Central Server (ACS) to manage *ARP table* entries in all hosts. In this technique all the host in the network uses the ACS to validate their *ARP table* entries. The ACS validates and corrects the poisoned ARP entries of the attacked hosts and hence prevents ARP poisoning in the network.

Seung Yeob Nam *et.al* [12] An enhanced version of Address Resolution Protocol (ARP) is proposed to prevent ARP poisoning based Man-in-the-Middle (MITM) attacks. They proposed a voting-based resolution mechanism to detect ARP Poisoning.

Ahmed M.AbdelSalam *et.al* [13] They proposed a scalable technique to prevent *ARP spoofing* attacks, which automatically configures static ARP entries. The technique based on both static and DHCP based addressing schemes, and technique protects large number of users without any overhead on the administrator. They conducted Performance study on a real network.

Jaideep Singh *et.al* [14] They study the lot a of measures that have been implemented to defend the spoofing attacks. They have concluded that the ideal method used to prevent ARP attacks is D-ARP, used by CISCO.

III. EXPERIMENT SETUP

Address Resolution Protocol being a stateless protocol, will cache any ARP replies received from the network hosts, regardless the request has been send or not. ARP protocol lacks authentication mechanism by which it can verify the authenticity of the sender of ARP reply packet. On receiving the ARP reply from any host in the network, *ARP table* is updated with the <IP, MAC> pair send in the *ARP reply* packet without verifying the source of ARP reply. Thus, the attacker can easily poison the *ARP cache* by spoofing the ARP replies. A variety of software are freely available to perform *ARP spoofing* attacks e.g. Ettercap[5], Cain and Abel[3] and dsniff[4].

In this experiment, a LAN of six PCs is setup to trace out *ARP Spoofing* vulnerabilities. In this experiment, host PC with IP address 192.170.1.120 and MAC address 64:31:50:3B:03:F2 acts as attacker and attack is performed on victim PC with IP address 192.170.1.121 and MAC address 94:57: A5:AC:29:E3. Ettercap tool is installed on attacker PC to produced spoofed packets.

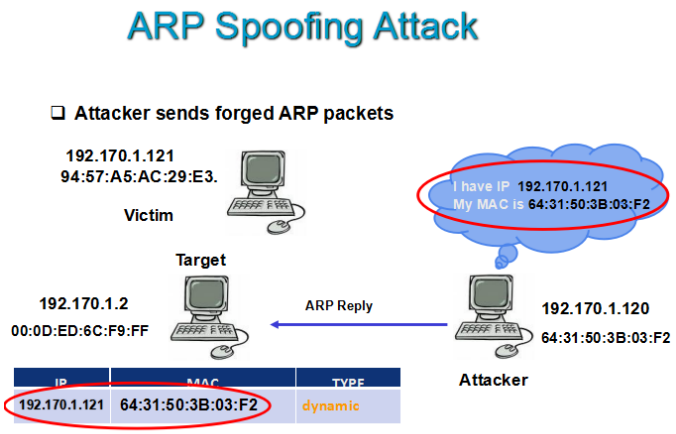


Figure 1 *ARP Spoofing*

In the figure 1 the attacker with IP 192.170.1.120 and MAC address 64:31:50:3B:03:F2 sends spoofed *ARP reply* packets to target with IP address 192.170.1.2 and MAC address 00:0D:ED:6C:F9:FF. Target's *ARP table* is spoofed by associating attacker's MAC address i.e. 64:31:50:3B:03:F2 with IP address of the victim i.e. 192.170.1.121. Attacker also spoof the *ARP table* of the victim by associating its MAC address with IP address of the target 192.170.1.2. In this way the communication between the target and the victim is controlled by the attacker and the attacker is now capable of performing a number of Man-in-the-Middle Attacks.

Snort has been installed in NIDS mode on target machine to detect spoofing. It captures and assesses all the ARP reply packets that the target receives to identify spoofed packets and also identifies the MAC address of the attacker machine.

IV OBSERVATIONS

In the experiment conducted, ARP Poisoning makes the network vulnerable for various severe attacks.

Cookie Stealing : When a website is visited from a browser, the data specific to client and website is sent in the form of cookies and store in the client's browser. The server uses this information to identify the given user in subsequent visits. By *ARP Spoofing*, the attacker can access the cookies directed for the target. In this experiment, cookies have been stolen and are analyzed by Wireshark packet analyzing tool installed on the attacker's machine. Figure IV(a) shows cookies of rediff.com intended for victim machine have been stolen and analyzed by the attacker.

Figure IV(a) cookies of rediff.com

Session Hijacking: Session Hijacking allows the attacker to take control of the connection between target and victim machines, and continues the connection with the target machine pretending to be the authenticate user. In this attack, session id of the victim is stolen, to gain the access of target session. In figure IV(b) we have stolen the session Id of the balesproject.com .

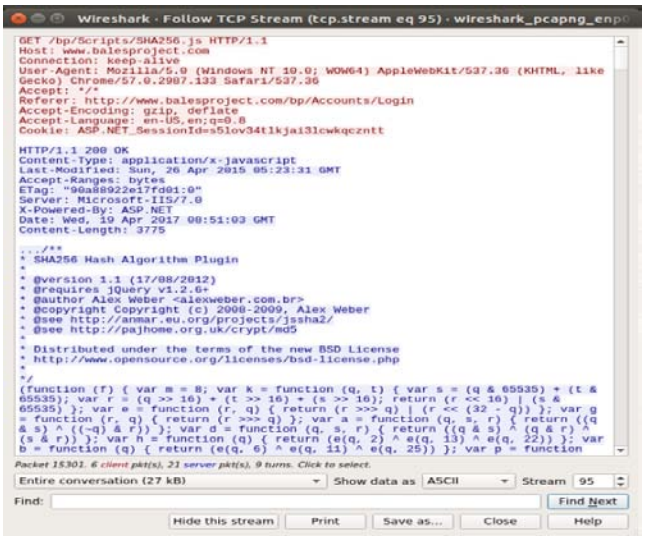


Figure IV(b) session id stolen by the attacker.

In this study, the network of six hosts is setup to observe the vulnerability associated with *ARP Spoofing* and performance study of the Snort for detecting *ARP Spoofing* has been conducted on a real network by varying number of the target from 1 to 5.

Detection

For detecting *ARP Spoofing*, Snort has been used and run in intrusion detection mode. Snort has been installed on one machine in the LAN to capture and analyze all the ARP packets in order to detect ARP Spoofed packets and generate alerts for them. For this, we have to make some changes in Snort preprocessor known as Arpspoof preprocessor[1] present in the Snort configuration file. We have to remove '#' before the following lines of code :

#preprocessor arp spoof
#preprocessor arpspoof detect_host 192.168.40.1 fO:Of:OO:fO:Of:OO

And modify them as:

preprocessor arpspoof
preprocessor arpspoof detect_host: host_ip host_mac

After modification in Snort configuration file, Snort detects *ARP Spoofing* by generating alerts for the spoof ARP replies. Alerts generated for *ARP spoofing*, are stored in alerts file.

```
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
05/18-10:12:34.896371
```

Figure IV(c) shows alert generated by snort.

The log file generated by snort when opened by Wireshark gives us the MAC address of the attacker.

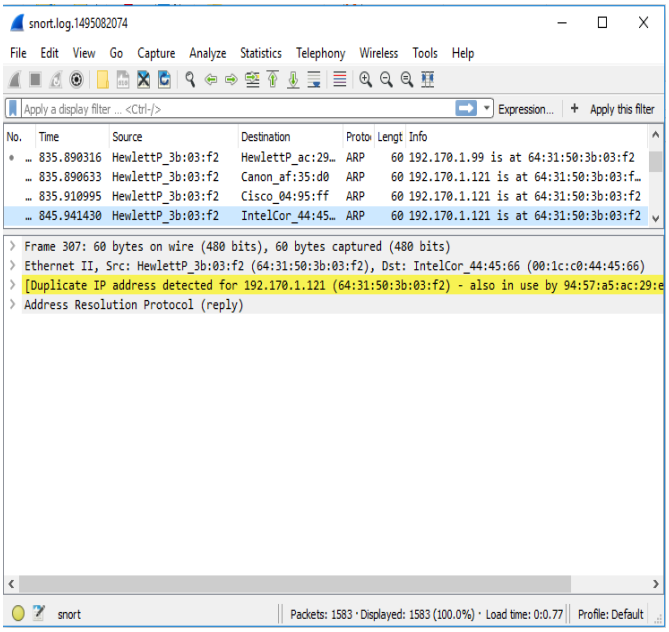


Figure IV(d) log file of snort opened in Wireshark

V RESULT

By varying the number of the victim from 1 to 5. *ARP Spoofing* is done for 1 hour, following results are observed.

Table 1 shows the number of ARP packet captured by sort

Number of targets	Number of alerts generated by Snort
1	1583
2	2626
3	5410
4	6443
5	8325

The table above shows the number of ARP packet captured by snort. With the increase in the number of hosts ,the number of packets received by snort are also increasing as shown in the graph.

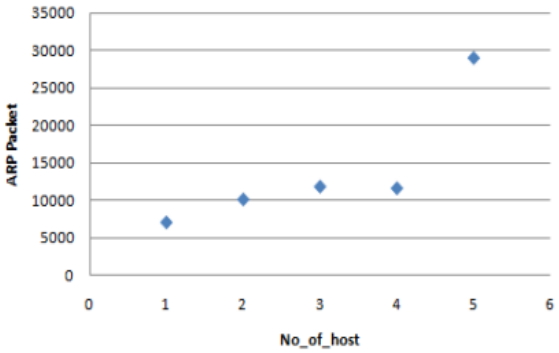


Figure V(a) shows the plotting of ARP packets captured by snort corresponding with number of victim.

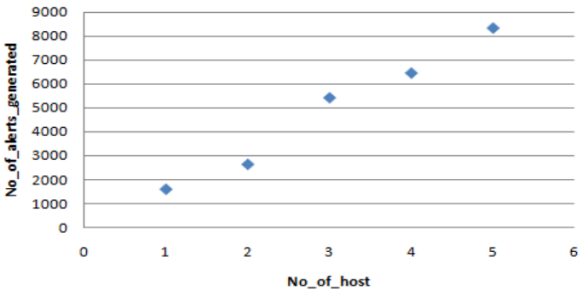
The table below shows the number of alerts generated by Snort for one to five hosts.

Table 2 shows the number of alerts generated by snort

Number of targets	Number of ARP packet captured by snort
1	7043
2	10102
3	11809
4	11589
5	28952

The following plot shows that the number of alerts are increasing with the increase in the number of spoofed hosts.

The two graphs show snort performance does not decrease by increasing number of the host being spoofed. It have been found that snort is scalable in detecting *ARP Spoofing*.



The figure V(b) shows the plotting of alerts generated by Snort corresponding with a number of victim.

VI CONCLUSION

ARP spoofing is the vulnerability which still exists in modern networking protocols. Because of the availability of number of automated *ARP Spoofing* tools, it is easier to implement and is frequently used by intruders to attack the network. We have used Snort to detect *ARP Spoofing* in the internal network. Snort detected the MAC address of the intruder in the network. This MAC address was blocked from sending any packets in the network. Snort detection was not affected by increase in the number of hosts present in the network.

REFRENCES

[1] http://manual-snort-org.

[2] Bruschi, D., Ornaghi, A. and E. Rosti, “S-arp: a secure address resolution protocol,” IEEE Conference on Computer Security Applications Conference, pp. 66 – 74, 2003.

[3] Cain & Abel, http://www.oxid.it/cain.html

[4] http://www.giac.org/paper/gsec/810/introduction-dsniff/101714

[5] A. Ornaghi, http://ettercap.sourceforge.net/

[6] M.Bamaba. “Anticap"http://www.antifork.org/svn/trunk/anticap

[7] http://www.oxid.it/cain.html I.Teterin, "Antidote" http://online.securityfocus.com/archive/l/299929

[8] Xing,W., Zhao,Y and Li,T., "Research on the defense against ARP spoofing attacks based on winpcap,” IEEE Second International Workshop on Education Technology and Computer Science, 2010

[9] Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670. (Article in conference proceedings)

[10] Vinay K.R and B.K. Gudur “ Detection of ARP spoofing attack using ICMP protocol,” International Journal of Computer Science and Mobile Computing, Vol. 3, Issue. 5, May 2014

[11] Sumit Kumar and Shashikala Tapaswi “ A centralized detection and prevention technique against ARP poisoning, “ Published in : Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference

[12] Seung Yeob Nam, Dongwon Kim, and Jeongeun Kim, “Enhanced ARP: preventing ARP poisoning-based Man-in-the-Middle attacks,”IEEE COMMUNICATIONS LETTERS, VOL. 14, NO. 2, FEBRUARY 2010

[13] Ahmed M.AbdelSalam ,Wail S.Elkilani ,Khalid M.Amin “An automated approach for preventing ARP spoofing attack using static ARP entries,”(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, 2014

[14] Jaideep Singh and Vinit Grewal “ A survey of different strategies to pacify ARP poisoning attacks in wireless Networks,” International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 11, April 2015