Comparative Investigation of ARP Poisoning Mitigation Techniques Using Standard Testbed for Wireless Networks

Goldendeep Kaur and Jyoteesh Malhotra

Computer Science and Engineering Department, Guru Nanak Dev University, Regional Campus, Jalandhar
E-mail: goldenchugh@gmail.com; jyoteesh@gmail.com

Received 4 September 2015; Accepted 25 September 2015;
Publication 22 January 2016

Abstract

Due to the increasing demand of wireless networks, there is an increasing necessity for security as well. This is because unlike wired networks, wireless networks can be easily hacked form outside the building if proper security measures are not in place as wireless networks make use of radio waves and radio waves can leak outside of building at distances up to 300 feet or more. So everything we do on our network can be monitored by anyone who has wireless capabilities. This unauthorized access can be used as an essence by the hacker to launch various kinds of attacks like man-in-the-middle attacks, denial of service attacks, IP spoofing etc. As a result in addition to the firewalls, password protection techniques, virus detectors etc, additional levels of security is needed to secure the wireless networks. This paper focuses on comparing various techniques that are used to protect the users from these attacks by providing practical observations based on the network parameters time and scalability and also highlighted the best method in the end to combat the attacks at a superior level.

Keywords: ARP Cache, Snort, Wireshark, SSLstrip, Ettercap.

Journal of Cyber Security, Vol. 4, 53–64.
doi: 10.13052/jcsm2245-1439.423

1 Introduction

Nowadays, Internet is a versatile facility for satisfying the people with various services related to various different fields. In this age of advancement of technologies, almost everything is now available over the internet that can help us in completing many tasks easily and conveniently with just few clicks whether it is any task of daily usage or some specific service which requires lot of research to be done beforehand. We can purchase anything and pay our bills online by going through various websites and choosing among variety of options. As a result, there might be some confidential data to purchase anything or pay online. So security should be taken into account. There are some mechanisms trying to provide secured information on the internet such as HTTPS, SSL etc. HTTPS was introduced to be used as a secure channel rather than HTTP connections. It provides reliable communication over the internet in terms of security. Almost every website today uses this protocol to run their business. However, one drawback of HTTPS is that it cannot tolerate SSL man in the middle attack that leads to security issue when the confidential information is hacked. Many researchers are working in this area to protect secure data by using cryptographic techniques, web browser plug-in and many more software's used to mitigate these attacks.

When a device needs to communicate with any other device on the same wireless network, it checks its ARP cache to find the MAC Address of the destination device. As a result of this check, if the MAC address is found in the cache, it is used for communication. If not found in local cache, the source machine generates an ARP request. The source broadcasts this request message to the local network. The message is received by each device on the LAN as a broadcast. ARP is a stateless protocol; therefore all client operating systems update their cache if a reply is received, inconsiderate of whether they have sent an actual or faked request. Since ARP does not offer any method for authenticating replies in the network, these replies are vulnerable to be manipulated by other hosts on a network. [3]

This paper provides the comparative analysis among the various defense strategies used to detect and mitigate these attacks based on the network para- meters time and scalability. It takes into account the experimental observations of each method and highlights the best method to mitigate these attacks.

The remainder section of this paper is organized as follows: Section 2 describes SSL MITM Background, Section 3 describes the experimenting

details, Section 4 describes the comparative analysis and Section 5 concludes the paper.

2 SSL MITM Background

Secure Socket Layer/ Transport Layer (SSL/TLS) [6] was originally developed by Netscape. It is used to deploy confidentiality, authenticity and integrity between web client and web server. SSL follow a standard handshake procedure to establish communication between client and server. The handshake prior to an HTTPS session follows:

1. The client contacts a server that hosts a secured URL.
2. The server responds to the client's request and sends the server's digital certificate (X.509) to the browser.
3. The client now verifies that the certificate is valid and correct. Certificates are issued by well-known authorities (e.g. Thawte or Verisign).
4. The server could optionally choose to confirm a user's identity. The TLS-enabled server software can check that the client's certificate is authorized and has been issued by a certificate authority (CA) from the server's list of trusted CAs. This confirmation is important in confidentiality as if the server is a bank sending confidential financial information to a customer and wants to check the recipient's identity.
5. Once the certificate is validated, random one-time session key is generated by the client to encrypt all communication with the server.
6. The client now encrypts the session key with the server's public key, which was transferred with the digital certificate. Encrypting using the server's public key assures that others cannot eavesdrop on this sensitive exchange.

In this way a secure session is established between the client and server, both knows the session key and can communicate via a secure channel.

However with the new intrusion tools developed, the SSL can be stripped apart. Moxie Marlinspike [9], during a BlackHat conference in 2009 released a tool known as SSLStrip which rendered the use of SSL digital certificates unfruitful. Here, the attacker created a fake digital certificate with spoofed identity and echoed the data both ways between client and the server. SSLStrip rewrites all HTTPS addresses as HTTP addresses and then saves traffic content. But these can have threatening effects on the organizations.

Security measures need to be taken to safeguard the interests of users. Various mechanisms that are used to mitigate these attacks are described in Section 3.

3 Defence Strategies Experimenting Details

One of the simplest ways to detect ARP poisoning is to list the ARP table which contains all the MAC addresses our computer knows. Now under ARP poisoning, there should necessarily be a duplicate MAC address. To detect it, open CMD as administrator and type the command:

"arp –d –a 3"

This command will clear the ARP cache i.e. it will clear possible disconnected devices on our network as shown in Figure 1.

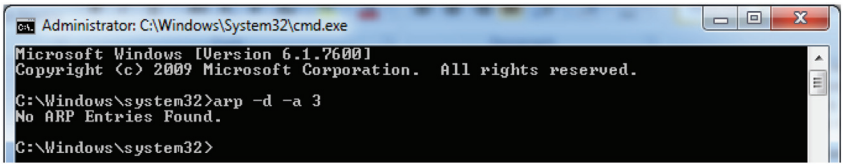

Figure 1 Clear ARP cache.

Now type the command:

"arp –a"

This command will list the ARP table and if there is any duplicate MAC address found in the ARP table. This is a clear indication of ARP poisoning.

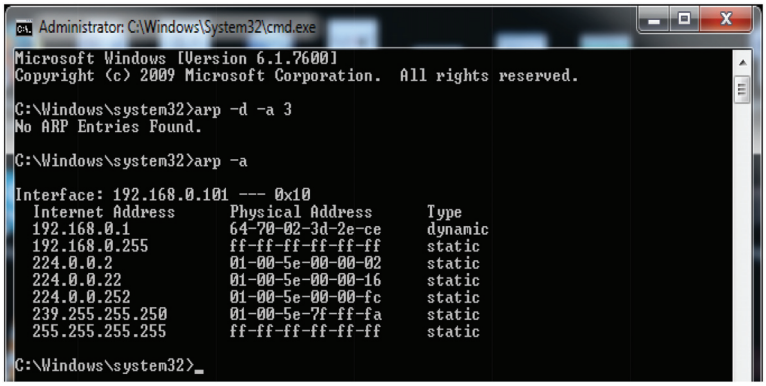

Figure 2 ARP table.

The disadvantage of this method is that it is not scalable for large networks as it would be very difficult to check each entry manually for duplicates.

Another method that is used to detect ARP poisoning is using Wireshark [7]. For the Wireshark to detect duplicate IP addresses, we need to capture the network traffic by selecting our network adapter from the interfaces list and click on start button to capture packets.

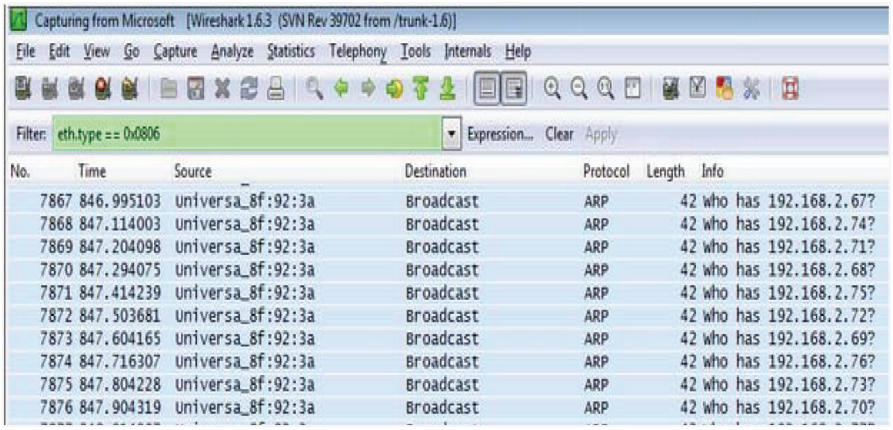

Figure 3 Normal traffic flow.

Now click on "edit preferences" from the top menu on right hand side as shown in Figure 4.

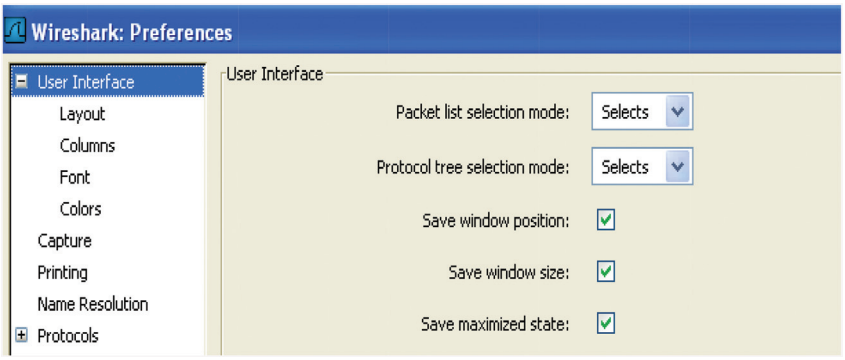

Figure 4 Edit preferences.

Select the protocol ARP/RARP from the Protocols tab and enable the feature "Detect duplicate IP address configuration" as in Figure 5.

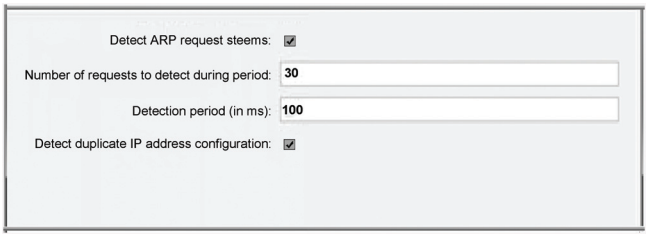

Figure 5 Enabling detection.

When this option is enabled, it will detect if anyone is trying to perform ARP poisoning on the network.

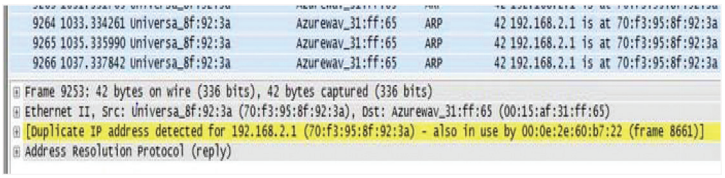

Figure 6 Duplicate IP address detected.

In this way Wireshark can be used to detect ARP poisoning on the network.

Ettercap [5] supports a variety of loadable modules at run time known as plug-ins. These plug-ins are automatically compiled until we specify disable plug-in option to the configure script. The available plug-ins in Ettercap is shown in Figure 7.

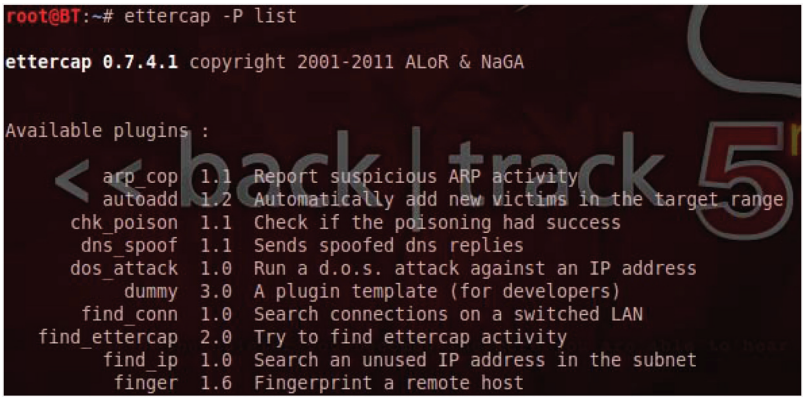

Figure 7 Ettercap plug-ins.

There is one plug-in "find_ettercap" that is used to find the Ettercap activity.

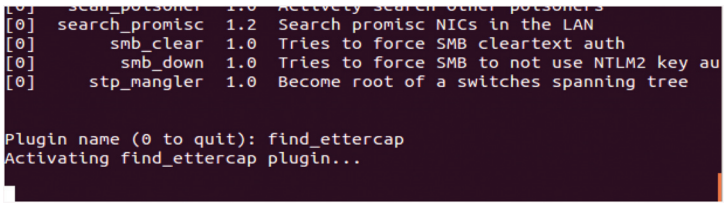

Figure 8 Find Ettercap activity.

The plug-in "search_promisc" is used to find if any machine is running in promiscuous mode.

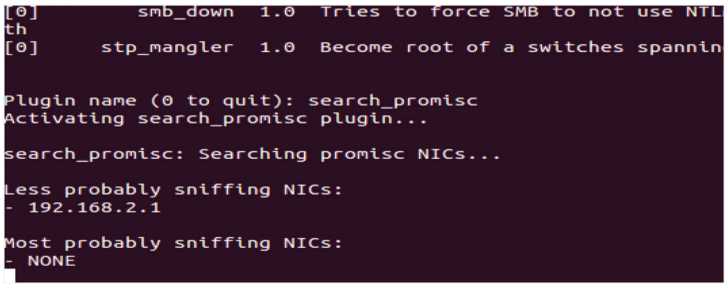

Figure 9 Searching promiscuous NIC.

Finally the "chk_poison" plug-in is used to check if anyone is trying to perform ARP poisoning on the network.

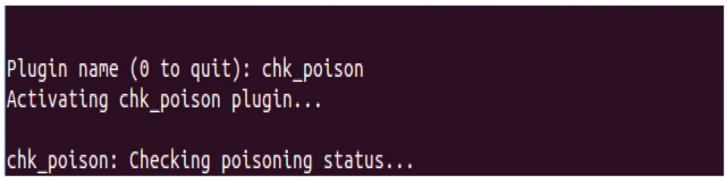

Figure 10 Checking ARP poisoning.

Intrusion Detection Systems like Snort [11] are available freely to detect any kind of attack. Snort is an open source system by Sourcefire. It is most widely deployed technology worldwide. Snort service is available in Backtrack. We need to start this service to monitor to start detection activity on the network.

Figure 11 Start snort.

We need to configure snort before using it. To configure it, we use vim editor and open the snort configuration file by the following command:

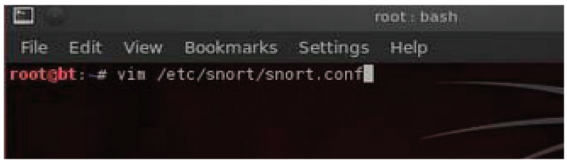

Figure 12 Opening snort configuration file.

In the Snort configuration file, we will see a line "var HOME_NET any". Change the word any to our own IP address as in Figure 13.

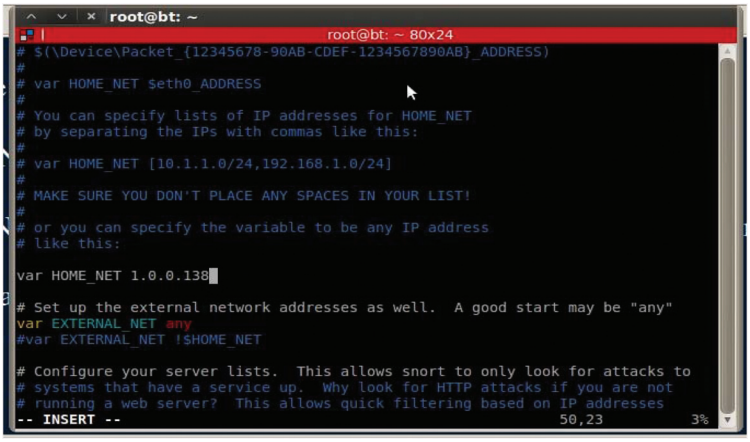

Figure 13 Configuring snort.conf file.

Now if anyone will try to perform any of kind of attack like man-in-the-middle attack, denial-of-service attack etc. Snort will detect it like in Figure 14.

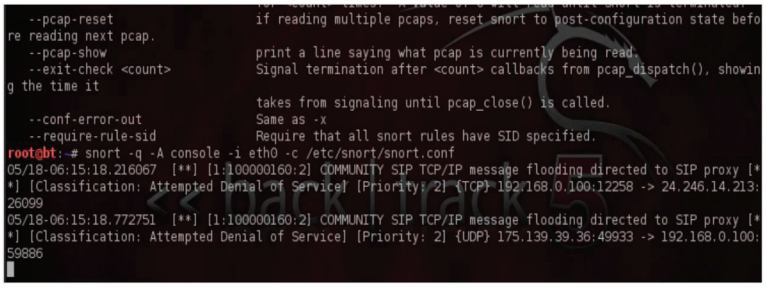

Figure 14 Denial of service attack detected.

In this way Snort can be used on Backtrack to detect any kind of attack.

4 Results and Analysis

To highlight the current state of art in this area, the most important and apt techniques have been tabulated in Table 1. The analysis reveals that if time is an important factor taken into consideration then detection using command prompt is least suited method to combat ARP poisoning assaults followed by Snort, then Ettercap plug-ins and then best method i.e. Wireshark. To support the view, a graphical analysis of various explored techniques has been given in Figure 15.

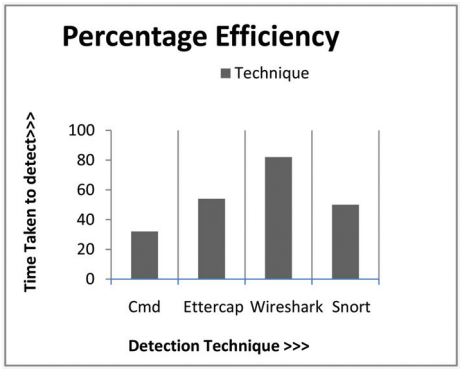

Figure 15 Bar graph analysis taking time into consideration.

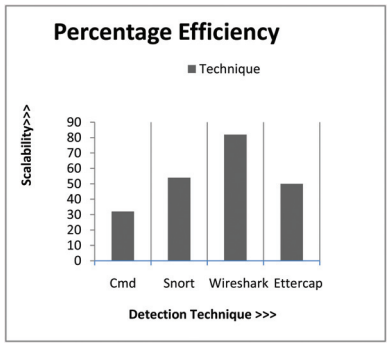

Figure 16 Bar Graph analysis taking scalability into consideration.

If scalability is an important factor taken into consideration then detection using command prompt is least suited method to combat ARP poisoning assaults followed by Snort, then Ettercap plug-ins and then best method i.e. Wireshark. To support the view, a graphical analysis of various explored techniques has been given in Figure 16.

5 Conclusion

It can be concluded that computer networks are vulnerable to dangerous attacks even today. Although every website today has preliminary protection on the information using HTTPS protocol instead of HTTP, still hacker is able to harvest the confidential information by using SSLstrip. He has to hook up to the same network with a victim and can monitor everything happening on the network. This paper as a result provided the analysis of best techniques to detect these attacks taking into consideration two network parameters time and scalability. The bar graph analysis shows that in both the cases Wireshark is the best method to deal with the attacks. In future, an ideal mechanism can also be developed and implemented by accomplishing several experiments with reference to the work done in wide scenarios of ARP spoofing attacks.

References

[1] "ARP-Guard," (accessed 28-July-2013). [Online]. Available: http://www.arp-guard.com.

[2] B. D. Schuymer, "ebtables: Ethernet bridge/switch tables," Mar. 2006, (accessed 28-July-2013). Available:http://ebtables.sourceforge.net.

[3] D. Plummer. An ethernet address resolution protocol, Nov. 2010. RFC 826.
[4] http://www.backtrack-linux.org/
[5] http://ettercap.sourceforge.net/
[6] https://www.digicert.com/ssl.htm
[7] https://www.wireshark.org/
[8] L. N. R. Group. arpwatch, the Ethernet monitor program; for keeping track of ethernet/ip address pairings. (Last accessed April 17, 2012).
[9] Moxie Marlinspike, "SSLStrip, Black Hat DC 2009", Retrieved http://www.thoughtcrime.org/software/sslstrip/
[10] S. Whalen, "An introduction to ARP spoofing," 2600: The Hacker Quarterly, vol. 18, no. 3, Fall 2001,. Available:http://servv89pn0aj.sn.sourcedns.com/_g bpprorg/ 2600/arp spoofing intro.pdf
[11] Snort Project, The Snort: The open source network intrusion detection system. <http://www.snort.org>.
[12] Demuth and A. Leitner. ARP spoofing and poisoning: Traffic tricks. *Linux Magazine*, 56:26–31, July 2011.