ARP Cache Poisoning Mitigation and Forensics Investigation

Heman Awang Mangut
ACE, UEL, United Kingdom
Email: u1345346@uel.ac.uk

Ameer Al-Nemrat
ACE, UEL, United Kingdom
Email: ameer@uel.ac.uk

Chafika Benzaïd
USTHB, Algérie
Email: cbenzaïd@usthb.dz

Abdel-Rahman H. Tawil
ACE, UEL, United Kingdom
Email: atawil@uel.ac.uk

Abstract—Address Resolution Protocol (ARP) cache spoofing or poisoning is an OSI layer 2 attack that exploits the statelessness vulnerability of the protocol to make network hosts susceptible to issues such as Man in the Middle attack, host impersonation, Denial of Service (DoS) and session hijacking.

In this paper, a quantitative research approach is used to propose forensic tools for capturing evidences and mitigating ARP cache poisoning. The baseline approach is adopted to validate the proposed tools. The evidences captured before attack are compared against evidences captured when the network is under attack in order to ascertain the validity of the proposed tools in capturing ARP cache spoofing evidences.

To mitigate the ARP poisoning attack, the security features DHCP Snooping and Dynamic ARP Inspection (DAI) are enabled and configured on a Cisco switch. The experimentation results showed the effectiveness of the proposed mitigation technique.

Keywords—ARP cache poisoning; Forensic investigation; Attack mitigation.

I. INTRODUCTION

Computer networks provide means through which users communicate, and make purchases through the Internet. Providing security to computer networks becomes imperative as the use of Internet in e-commerce and communication increases, proliferation of computer network threats is also spreading fast [1]. There are a variety of computer network threats operating at different layers of computer network [2]. These threats include DoS, DDoS, ARP cache spoofing, DNS poisoning, viruses etc. [3].

Controls needed for protecting computers from threats or attacks are dependent on decision variables that the management of an organization makes [4]. When a particular threat becomes a concern, controls needed for mitigating such such attacks should be be convergent [5]. Hence, to prevent ARP cache spoofing threats on the network, decision tools such as baseline approach or risk assessment approach need to be adopted to ascertain that all controls needed to mitigate such attacks as part of organization’s Business Impact Analysis (BIA) are considered [6].

Some organizations make Business Impact Analysis (BIA) for controls to mitigate attacks on their information assets. Usually, they focus on external attacks coming from outside their network, while neglecting internal attacks launched from inside the network. In some other settings, an organization could have controls in place, however it may neglect the fact that a computer network is considered secure only if the vulnerability in it has not yet been discovered. Hence, this type of organizations may fail to put forensics tools in place against unknown attacks. It is important that they are able to study those attacks which could help them forestall feature occurrence and possibly identify source of attacks whether internal to the organization or external [7]. ARP cache spoofing is the commonest attack that can be launched from within a network and could have a very high destruction profile to an organization.

The purpose of this research work is to use baseline approach to validate the proposed forensics and mitigation approach against; (1) ARP cache spoofing, (2) Man in the Middle attack, (3) Host impersonation, (4) Denial of Service attack (DoS), (5) Session hijacking.

The rest of the paper is organized as follows. Section II gives an overview on attacks based on ARP cache poisoning. Section III summarizes related work in the literature. Section IV describes the investigation and mitigation methodology adopted. Section V presents the forensic investigation approach, followed by a detailed analysis of validation results. The mitigation approach is presented in Section VI. Finally, Section VII concludes the paper.

II. OVERVIEW OF INTERNAL ATTACKS BASED ON ARP CACHE POISONING

A. Man in the Middle Attack

Man in the Middle (MitM) attack is a hacking methodology whereby an attacker poisoned the ARP caches of two communicating hosts to intercept their communication with the aim of causing host exploitation such as session hijacking, theft of sensitive data, port stealing and impersonation of login credential [8]. To launch the attack, the attacker first collect the MAC addresses of its victims by broadcasting an ARP request to the victims’ entire network. After that, the attacker sends an ARP reply to the victim hosts in order to associate their IP addresses to its MAC address.

B. Denial of Service (DoS) Attack

According to [9], DoS is an exploitation of ARP cache poisoning by an attacker to identify itself as the default gateway to the victim host. Thus, all traffic sent to the
gateway will be redirected to the attacker which will choose to drop. The attacker can also give the victim host a fake default gateway which does not exist on the network. Hence, the victim host will lose connection to the network.

C. Host Impersonation

An adversary can exploit ARP cache poisoning to impersonate another host and gain access to sensitive information sent to this host [10].

D. Session Hijacking

Session hijacking consists in taking over an active communication session of a legitimate user (i.e., victim) once it has authenticated to a server [11].

III. RELATED WORK

Gouda and Huang [12] proposed a mitigation technique for ARP cache poisoning using invite-accept protocol and request-reply protocol such that host computers on the network could have their IP-MAC address binding stored on a database of a centralized server on the network. A host computer uses the protocols to get any host it needs to communicate with on the network. However, [13] argued that the implementation of this technique requires the alteration of ARP protocol of every. Moreover, the server could be a single point of failure and proposed mitigation technique is prone to DoS attack.

A Spoofing Prevention Method (SPM) is proposed in [14], where a distinctive key is associated with the source and destination domains. The key is appended into a packet at the source end and verified at the destination end.

[15] proposed the use of Hash Message Authentication Code (HMAC) to check the integrity and authenticity of user cookies. HMAC is computed on four concatenated variables: username, expiration time, data, and session key. The cookies are stored on the server side which complicates the possibility of session hijacking to an attacker. However, the proposed protocol is not scalable due to its reliance on SSL. In a related development, [16] proposed SessionLock; a mitigation technique against session hijacking. To this end, SessionLock relies on a session secret to generate unique authentication tokens and uses HMAC. Also, [17] proposed SessionShield to prevent session hijacking. SessionShield is a proxy outside of the browser that inspects all outgoing requests and incoming responses to identify session cookies, and then blocks script access to these session cookies. Nevertheless, the use of a proxy technique poses a scalability and compatibility issues [18].

Although efforts have been made to provide mitigation techniques to ARP cache spoofing, each proposition has its weaknesses which makes the quest for an ARP cache spoofing mitigation lingering. Besides, none of the propositions presented above have addressed the attack issue from forensics point of view. Whenever there is an attack on a network, there is always a need to provide evidence which is conform to forensics investigation standard. This allow to infer that the attack has really happened and to be able to identify the source of the attack so that it could be used against the suspect in the law court [19].

IV. THE PROPOSED INVESTIGATION AND MITIGATION METHODOLOGY

In this work, a qualitative approach is used to collect data. A baseline approach is adopted to verify whether the network is under attack or not using several forensic tools, namely: TCPdump [20], Wireshark [21], and Linux commands TCPstat and Ntop. A baseline is a “value or profile of a performance metric against which changes in the performance metric can be usefully compared” [6]. In our case, the evidences captured before attack are compared against evidences captured when the network is under attack.

To mitigate the ARP cache poisoning attack, DHCP snooping and dynamic ARP inspection (DAI), two port security features, are enabled and configured on the switch. Those security features are available on the latest Cisco Catalyst switches, including 6500, 4500 and 2960 Series. The Cisco Catalyst 2960 Series switch is a reasonable choice for Small and Medium Enterprises (SMEs) due to its affordable price. To validate the proposed mitigation technique, an experiment network is setup as shown in Figure 1.

The network comprises a Cisco router 1841, a Cisco Catalyst 2960 switch, a victim machine running Windows 7, and a laptop containing two virtual machines (VMs) running Linux family Operating Systems (OS) on Windows 8.1 professional. The VM running Kali Linux serves as the attacker host. Two Ethernet-to-USB adapters are used to connect each of the VMs running on the laptop to the external DHCP in order to ensure that all hosts are connected to the same network. The network 192.168.50.0/24 is used with the default gateway 192.168.50.1 configured on Fast Ethernet 0. DHCP is enabled on the router. VLAN 50 is configured on the switch. Fast Ethernet 2, 3, 5, 7 and Gigabit Ethernet 1 interfaces are assigned to VLAN 50. All devices are put under VLAN 50 to avoid possibility of interference on the result as a requirement for a valid research design.

To validate the proposed forensics investigation of ARP cache spoofing, comparison will be made between the
forensics evidences collected when there is no attack on the network against those collected when the network is under attack. Ettercap is used to launch the ARP cache poisoning and the corresponding attacks: MitM, DoS, session hijacking, and host impersonation. To validate the proposed mitigation technique, the evidences collected when the network is under attack with DHCP snooping and DAI enabled will be compared to those when the two security features are disabled.

V. FORENSICS INVESTIGATION OF ARP CACHE POISONING

A. Normal Traffic Flow on the Network

The first evidences were captured when the network was in normal operation (i.e., no attack) using TCPdump, Wireshark, TCPstat and Ntop. The extracted evidences will be used as a baseline to ascertain whether the network is under attack or not.

![Figure 2. TCPdump output when the network traffic was normal](image)

Figure 2 shows the packets collected with TCPdump. The TCPdump output comprises eleven fields that represent, respectively: the timestamp, the protocol, the source IP address and port, the destination IP address and port, the TCP flags, the relative sequence number, the relative acknowledgement number, the window size, and the packet length. For instance, the first line says that an IP packet was sent at timestamp 23:03:07 from source port https on the source host 74.125.206.18 to destination port 49252 on the destination host (in our case, the victim machine) 192.168.50.5. The https denotes that response from source host was over a Secured Socket Layer (SSL/TLS). The TCP flags F and P are set in the packet. The FIN flag, F, indicates the sender's intention to terminate the connection to the receiving host. The PUSH flag, P, signals the immediate push of data from the source host to the destination host. The packet's relative sequence number was 0 (seq 0) and it contained no data (length 0). A sequence number of zero indicates that this is the first packet sent by the source in this TCP session. A packet length of zero means that the packet contains only the header without payload. The available receive window was 64240 bytes.

The evidences captured by Wireshark are described in Figure 3. Only ARP traffic is displayed. The results show that the victim machine with MAC address Dell_31:ba:fd and IP address 192.168.50.5 made a broadcast asking for the MAC address of the default gateway with the IP address 192.168.50.1. It went further to making gratuitous ARP request of the existing default gateway. When a host’s IP address changes, a gratuitous ARP packet is transmitted on the network to force any device that receives it to update its cache with the new IP-to-MAC address mapping.

![Figure 3. Wireshark evidence output when traffic was normal](image)

![Figure 4. Output of evidence captured with TCPstate](image)

TCPstat is a network probing application which functions as an IDS. The traffic statistics depicted in Figure 4 report the estimated number of bytes transmitted per second, minute, hour, day, and month.

![Figure 5. Percentage of Unicast, Broadcast and Multicast packets captured by Ntop on the network when there was no attack](image)

Figure 5 shows the captured evidences using Ntop. It is observed that Ntop was activated on Ethernet port 1 (eth1) operating on a normal Maximum Transmission Unit (MTU) of 1514. The evidences were captured in 1 hour, 41 minutes, and 21 seconds. Neither libcap nor ntop have dropped any packet. Ntop received 6,123 packets that were processed. The total number (resp. percentage) of unicast, broadcast, and multicast packets was 2,364 (39.7%), 3,034 (49.6%), and 718 (11.7%), respectively.

B. Launching an ARP Poisoning Attack

The purpose of launching attack is to find the performance profile of the attack which will be used to measure the mitigation capability of DAI and DHCP Snooping on ARP cache snooping.

Ettercap is used to launch the attack. A unified sniffing is started on the Ethernet interface connecting the attacker machine to the network. The interface is placed into promiscuous mode. The default gateway and the victim machine are selected as the first and the second target machine, respectively. A MitM ARP poisoning attack is then initiated to spoof the ARP tables of the target machines.
From this point, the plugins "repoison_arp" and "remote_browser" are selected alternately to launch a session hijacking attack.

C. Evidence Collected under the Attack

From Figure 6, it could be seen that the time stamp was 23 hours, 17 minutes, and 1 second of the day. The IP source was from the subnet 74.125.206.0 transmitting over SSL, and the destination IP address was 192.168.50.7. The sequence number was 0, the acknowledgement was 1, length mostly 0, and the TCP flag was [FP] meaning “finish push”. Although, there was ARP Reply in two places with the IP address of the default gateway and that of the victim machine. The length of ARP reply packet is 46 bytes.

From the Ntop statistics shown in Figure 9, it could be seen that there was another ARP reply claiming to have originated from the first target (i.e., default gateway) with MAC address 00:0C:29:00:4f:2d but also with the attacker’s MAC address 31:ba:fd:08:00:01. Hence, something suspicious must be going on here. This demonstrate that the MitM by ARP poisoning attack triggered by ettercap was successfully launched causing the ARP cache poisoning for both targets.

D. Discussion

1) Analysis of Evidence Collected with Tcpdump: After comparing the timestamps of the captured evidences with Tcpdump when no attack was launched (timestamp was 23h3mn7s) and when the network was under attack (timestamp was 23h17mn1s), it could be seen that the evidence of a normal traffic flow was captured before launching the attack on the victim machine.

The IP address used by the source machine transmitted from subnet 74.125.206.0 over the Internet was on HTTPS in both cases. The source machine does not seem to maintain a unique IP address; its IP address appeared to be changing in fraction of a second. The victim computer’s IP address the suffixed port number remained the same. It is also observed that the TCP flags, the packet sequence and acknowledgement numbers, the packets’ length, and the window size remained the same in both cases. However, line 7 of evidence collected when the attack was launched shows that there was an ARP reply claiming to have originated from the second target (i.e., victim machine) with IP address 192.168.50.7 but with MAC address of 00 : 0C : 29 : 00 : 4f : 2d which was the MAC address of the attacker machine. In the same way, line 8 shows another ARP reply claiming to have originated from the first target (i.e., default gateway) with IP address 192.168.50.1 but also with the attacker’s MAC address 00 : 0C : 29 : 00 : 4f : 2d. Hence, something suspicious must be going on here. This demonstrate that the MitM by ARP poisoning attack triggered by ettercap was successfully launched causing the ARP cache poisoning for both targets.

2) Analysis of Evidences Collected with Wireshark: From the conversation depicted in Figure 10, it could be seen that the victim machine was making ARP broadcast asking for the default gateway. It is also observed that the victim machine was a dell product from the address column of the statistics table with shorten form of MAC address 31:ba:fd and the cisco switch too having shorten form MAC address of 29:2c:03.

Recall that the victim machine has made a gratuitous ARP request of the existing default gateway. The Figure 11 shows of unicast, broadcast, and multicast packets was 1,118,923 (99.8%), 794 (0.1%), and 1,080 (0.1%), respectively.
Figure 10. Conversation between the switch and the victim machine at that time. The conversation infers that the switch had acknowledged the existence of the victim machine, added the MAC-IP address binding to its ARP cache, and made a broadcast for the default gateway too.

Figure 11. Switch joined broadcast for the default gateway

From the evidence collected with Wireshark when the victim machine was under attack, we observed that from the “info” column it was reported “duplicate use of 192.168.50.5 detected”. The Figure 12 shows statistics of conversation at the time. It could be seen that another MAC address (34 : 78 : 74) was captured by Wireshark and a duplication of IP address was reported as shown in Figure 13. It could be observed that the IP address 192.168.50.1 which was earlier known to be the IP address of the default gateway has two MAC address bindings: 58 : 6d : 8f : 93 : 10 : 2e which is the initial MAC address of default gateway and 00 : 26 : 0b : 34 : 78 : 74 which is a fake MAC address of a host that does not exist on the network which was the outcome of the repoison plugins from ettercap. This effect should consequently result in DoS attack preventing access to the default gateway.

Figure 12. Conversation when victim computer was under attack

3) Analysis of TCPstat Collected Evidences: From the TCPstat statistics it could be obviously seen that the estimated traffic rate has increased when the network was under attack. This increase could be due to increase in ARP replies the repoison plugin of the ettercap was enabled.

4) Analysis of Evidences Captured with Ntop: It is obvious to see that the number of unicast during attack increased abruptly. This could be due to spoofed ARP reply from the attacker machine to the victim machine.

VI. MITIGATING ARP CACHE SPOOFING

In order to mitigate the ARP cache spoofing, the security features DHCP Snooping and Dynamic ARP Inspection (DAI) were enabled on the switch. The DHCP Snooping protects the network by allowing the switch to accept DHCP response message only from the authorized servers connected to the trusted interfaces in the switch. DAI helps preventing ARP poisoning and other ARP-based attacks by intercepting and verifying the authenticity of any ARP or ICMP request/reply, and dropping any ARP spoofing that is beyond the rate-limit configured on untrusted ports (in our case, this is the port on which the attacker machine is connected).

In our case, the two features were enabled on VLAN 50 and the Gigabit Ethernet port 1 (gi1/1) of the switch was configured as a trusted port. The DAI logging buffer size is set to 1024.

A. Launching Attack Again

At this time both DHCP Snooping and DAI security features were activated and configured on the switch. Then the ARP cache spoofing was launched again with ettercap as described in subsection V-B.

B. Analysis of Mitigation of ARP Cache Spoofing

Obtained results showed that the ARP cache spoofing was prevented when DHCP snooping and DAI were enabled and configured on the switch. Although both repoison_arp and remote_browser plugins were selected, session hijacking was not possible as when there was no mitigation. As the DHCP Snooping and DAI were configured to trust only ARP request/reply from the port connected to the DHCP router, the fake ARP request/reply may certainly be dropped by DAI. Note that we deliberately didn’t configure the rate limit of incoming ARP packets on untrusted ports in order to avoid putting the port in an error_disable state which could interfere with the size of data to be collected in the course of forensics investigation, if there could be any.

VII. CONCLUSION

The primary motivation of this research was to discuss the likelihood and to establish a baseline approach may used to mitigate and investigate ARP cache spoofing. Man in the Middle, host impersonation, Denial of Service, and
session hijacking attacks. The result obtained in this research showed that mitigation of ARP cache spoofing using DHCP Snooping and DAI was achievable. This research also introduced TCPdump, wireshark, TCPstat, and Ntop as a useful tools and techniques to be used in forensics investigation of ARP cache spoofing attack.

REFERENCES


