

SSL Stripping Technique (DHCP Snooping and ARP Spoofing Inspection)

Hannah A. S. Adjei*, Mr Tan Shunhua*, George K. Agordzo**, Yangyang Li*,
Gregory Peprah***, Emmanuel S. A. Gyarteng*

*College of Communication and Information Engineering, Southwest University of Science and Technology Mianyang, China

**School of Information and Electronic Engineering, Zhejiang University of Science and Technology Hangzhou, China.

***College of Engineering, Kwame Nkrumah University of Science and Technology Kumasi, Ghana

serwaa201479@gmail.com, 18298684@qq.com, k.agordzo@gmail.com,
w703710691d@163.com , Kpeprah1@gmail.com , e.sarp89@gmail.com

Abstract — Detecting and preventing network attacks are necessary for network solidity as any disruption in the network performance resulting from an attack can lead to loss of resources. Though ARP plays a vital role in successful local area network (LAN) communication, its vulnerabilities are used by attackers every day and by far have made it the leading point for refined LAN attacks such as; denial-of-service (DoS) and man-in-the-middle (MITM). This paper proposes a technique to mitigate an Address Resolution Protocol (ARP) spoofing-based Secure Socket Layer Stripping (SSL) in a Local Area Network (LAN) using both dynamic host configuration protocol (DHCP) snooping and ARP inspection technique. This mitigation technique includes a detection and prevention module. The detection module uses an analyzing tool and an algorithm that captures ARP packets. It identifies suspicious activity in the network, and once detected, a DHCP snooping, and ARP inspection technique is used to mitigate the attack to achieve a more secure network.

Keywords - DoS, LAN, ARP spoofing, MITM, DHCP Snooping

1.0 INTRODUCTION

ARP is a Data link layer communication protocol that maps an internet Protocol(IP) address to Media Access Address (MAC) by broadcasting an ARP request to the network and the desired host whose MAC address is required, sends a unicast reply with it's MAC address[1][2][3]. This mechanism is required because a switch network depends on a physical address for delivery and due to that, clients are required to have an updated table of logical-to-physical address bindings at all times. Because ARP is needed for successful communication in a LAN network, its vulnerabilities are exploited by attackers who use it to launch attacks such as DoS, secure socket layer (SSL) stripping, MAC flooding, and other attacks upon the success of becoming the man in the middle through ARP spoofing. When an attacker becomes a MITM, he can intercept packets that contain sensitive information such as the password and usernames, codes, and other sensitive information that the victim enters on websites or when using protocols that use readable texts such as

hypertext transfer protocol (HTTP), and teletype network (Telnet). He can modify captured packets before re-routing them to the victim machine [4], and can also inject malicious javascript in data packets. A man in the middle can downgrade most hypertext transfer protocol secure (HTTPS) websites to HTTP so that the attacker can access all information entered on the website by the victim [5].

1.1 Background

ARP spoofing is possible because; 1. Clients accept responses though they did not send a request. 2. Client trust responses without any form of verification [3][6][7][8]. There are two steps required for a successful ARP spoofing. These are;

1.1 Deceiving the Gateway

ARP packets are forged from the attacker machine to the gateway using the IP address of the victim and the MAC address of the attacker. Attacker fools gateway that it is the victim machine and upon that success, gateway updates its cache entry with the MAC address of the attacker and IP address of the victim because communication is done using MAC addresses in LAN, the gateway will now send data to the attacker machine who then chooses to modify and forward the packet to it's intended destination or drop it.

1.2 Deceiving the Victim Machine

The same process used in fooling the gateway is used here. The attacker sends an ARP reply packet to the victim telling it that it is the gateway by using the IP address of the gateway and the attacker's MAC address. Like the gateway, the victim updates it's cache table and now sends data meant for the gateway to the attacker unknowingly.

Steps needed for a successful MITM attack based on ARP spoofing:

First, the attacker must successfully come between two legitimate hosts to break the direct communication point between those two hosts with the attacker being in the middle of them. Upon success, data between these hosts are intercepted by the attacker in the middle.

Table 1:
Common Items and Its Unit Used throughout the Paper

Item	Unit
Internet Protocol	IP
Media Access Address	MAC
Local Area Network	LAN
Man-in-the-Middle	MITM
Denial of Service	DoS
Address Resolution Protocol	ARP
Dynamic Host Configuration Protocol	DHCP
Hypertext Transfer Protocol Secure	HTTPS
Hypertext Transfer Protocol	HTTP
Graphical Network Simulator-3	GNS3
Virtual Local Area Network	VLAN
Transport Layer Security	TLS
Secure Socket Layer	SSL
Dynamic ARP Inspection	DAI

2.0 RELATED WORKS

There are many proposed solutions in detecting and preventing ARP attacks. These solutions can be grouped into three:

Algorithms Detection Techniques: [3] proposed a detection algorithm using ICMP. It detects a malicious host attack. Although this solution effectively detects the attacker, no solutions were provided to prevent the attack. Ming [9] also proposed a detection algorithm for ARP attacks using data packet forwarding behavior but the downside of this technique is that there is no prevention technique after detecting the attack.

Tools and patches based solutions: There are several of these tools and patches used to detect and prevent ARP spoofing. Snort has been invented to detect MITM by [10]. This technique depends on static <IP, MAC> therefore, it becomes ineffective when there isn't a static entry saved between two hosts. In a Linux environment, a patch called Anticap is used [11]. However, usage is limited to only a few operating systems and also, works well only in a static environment. Antidote [12] is also used to parse the MAC address in the ARP response packet comparing it to the ARP cache entry. But, this technique is limited because it supports some kernels only.

Cryptographic Solution: S-ARP [13] uses the signature technology that includes a digital certificate and Authoritative Key Distributor (AKD) to modify ARP packets, and its downside is, it is forward incompatible and has a high computational cost.

2.1 The Proposed Solution

The following section describes in-depth the solution to ARP spoofing. This section is into three. The first part is the set up of the Virtual LAN (VLAN) using graphical network simulator-3 (GNS3), where a man in the middle attack is successfully launched on a windows machine by Kali Linux.

After becoming the man in the middle, Kali can view passwords, user names, and other sensitive information entered on the windows machine when clear text protocols such as HTTP and Telnet are used. HTTPS sites are downgraded to HTTP by performing SSL strip on the sites for readable information.

The second section is the detection mechanism of the attack launched in the first section. In this section, a detection algorithm is created using Winpcap in C to filter and capture all suspicious incoming ARP reply packets. Aside from the Algorithm that captures the packets, a powerful packet analyzer tool known as Wireshark is used to analyze the packets for suspicious activities. The tool in addition to the algorithm provides detailed information on the attack.

The last section is the prevention technique which uses combined DHCP snooping and ARP inspection techniques to prevent a successful man in the middle attack that leads to SSL stripping.

2.2 Contributions of this Study

The main contributions of this study are following:

- this work present a comprehensive experimental approach that explains in detail how to launch, detect, and prevent an ARP attack in a LAN as most researchers only do at most two of the following.
- It also show that different software and techniques combined can effectively prevent more than one attack. Because the same technique can be used to prevent some common LAN attacks such as ping of death (PoD).
- Both detection software and prevention techniques used are compatible with most operating systems and can be used in different environments.

3.0 EXPERIMENT SIMULATION NETWORK

The components of the experiments are listed here (see Fig. 1):

- GNS3 Graphical User Interface(GUI)
- VMware workstation player Ver14.0.0
- GNS3 VM on VMware
- Cisco IOSv 15.6(2)T Switch
- Cisco c3725 router
- Kali Linux Qemu Appliance in GNS3
- Windows 10 Qemu Appliance
- Wireshark packet capture in GNS3 on the port of victim(Windows) machine
- Program used to capture and analyze ARP reply packets based on WinPcap technique [14].

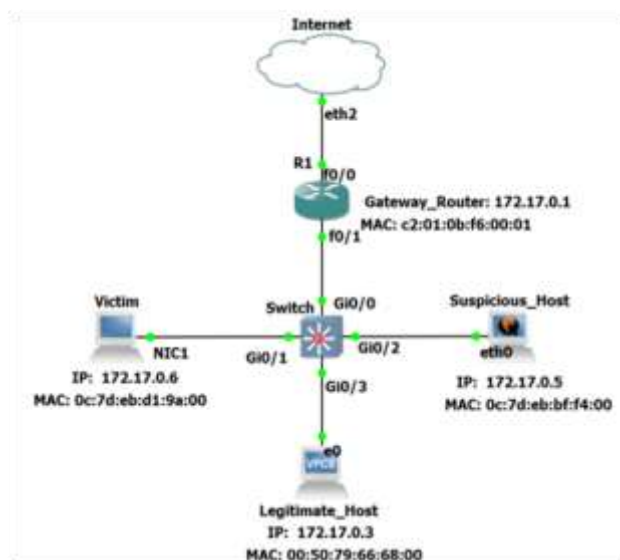


Figure 1: The Experiment Simulation Network Structure

Fig. 1 shows the topology of the experiment network. The attacker machine (IP:172.17.0.5) uses bettercap software to launch the ARP spoofing attack. A custom caplet is created and opened in a bettercap. This caplet contains the necessary commands needed to run the ARP spoofing attack to become the MITM. In Fig 2. The command “bettercap -iface eth0 -caplet spoof.cap” is used to run the ARP spoofing caplet (spoof.cap) created. The interface (iface) on which the attack launched is eth0. The command below opens and successfully runs the caplet (spoof.cap).

```
File Edit View Search Terminal Help
root@kali:~# bettercap -iface eth0 -caplet spoof.cap
bettercap v2.28 (built for linux amd64 with go1.14.4) [type 'help' for a list of
commands]
[09:18:20] [sys.log] [inf] [net.probe] starting net
[09:18:20] [sys.log] [inf] [arp.spoof] enabling for
[09:18:20] [sys.log] [inf] [arp.spoof] arp spoofer
[09:18:20] [sys.log] [inf] [arp.spoof] started, probing 1 targets.
[09:18:20] [endpoint.new] endpoint 172.17.0.6 det
```

Figure 2: Launching of Attack

Fig.3 and Fig.4 show the ARP table of the victim machine before and after the attack is launched. Comparing the two figures, one can see that the gateway <IP, MAC> before the attack is different from the <IP, MAC> after the attack. One can also see that the MAC address of the gateway (0c-7d-eb-bf-f4-00) now matches the MAC address of the attacker (0c-7d-eb-bf-f4-00) in fig. 3. This shows that the attacker has successfully fooled the victim into thinking that he is the gateway and has achieved his goal of becoming the man-in-the-middle.

```
C:\Users\IEUser>arp -a
```

Interface: 172.17.0.6 --- 0x6

Internet Address	Physical Address	Type
172.17.0.1	c2-01-0b-f6-00-01	dynamic
172.17.0.5	0c-7d-eb-bf-f4-00	dynamic
172.17.0.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static

Figure 3: ARP table of victim before attack

```
C:\Users\IEUser>arp -a
```

Interface: 172.17.0.6 --- 0x6			
Internet Address	Physical Address	Type	
172.17.0.1	0c-7d-eb-bf-f4-00	dynamic	
172.17.0.3	00-50-79-66-68-00	dynamic	
172.17.0.5	0c-7d-eb-bf-f4-00	dynamic	
224.0.0.22	01-00-5e-00-00-16	static	

Figure 4: ARP table of Victim after attack

Hypertext transfer protocol commonly known as HTTP is a communication protocol in the upper layer of the open system interconnection (OSI) model used for communication between a web-server and a browser. Because data in HTTP is sent in plain text, sensitive information can be easily accessed when this protocol is used and as such is highly not recommended for users to enter sensitive details on websites using HTTP[5][15].

After successfully becoming the man-in-the-middle, the attacker can capture passwords and usernames on the victim's machine when clear text protocols such as HTTP are used by websites. As seen in fig. 5, the victim has signed in on a website without knowing that the information he entered to log in has been captured by the attacker on his machine as shown in fig. 6. This shows that the attacker has successfully fooled the victim into thinking that he is the gateway and has achieved his goal of becoming the man-in-the-middle.



Figure 5: HTTP Site

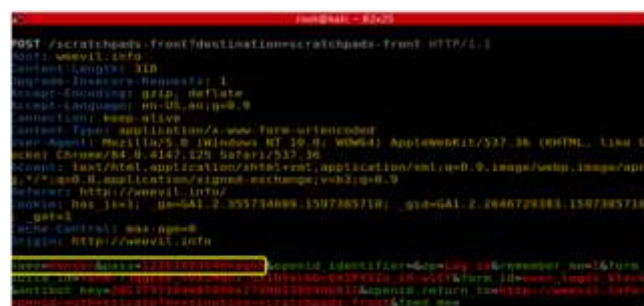


Figure 6: Captured Name and Password

Downgrading HTTPS to HTTP (SSL Stripping) hypertext transfer protocol secure (HTTPS) is an adaptation of HTTP, it is encryption of HTTP using a transport layer security (TLS) or secure socket layer (SSL). HTTP with an extra secured layer added to it becomes HTTPS. The function of this extra layer is to encrypt the plain data that HTTP [5][16] sends so that, even if an attacker becomes a man-in-the-middle, the data over HTTPS becomes gibberish to him as he is unable to read and understand the data. Therefore to bypass this

problem, the attacker downgrades HTTPS connections to HTTP by stripping the extra layer that encrypts the data so that if the victim requests for HTTPS websites, the attacker being the man in the middle gives HTTP version of the website instead. This way, the data will be sent in plain text, readable to the attacker. We use bettercap which has a default SSL strip caplet however, the default SSL strip caplet does not replace all HTTPS links to HTTP in the loaded pages so we modify the caplet to replace all HTTPS links to HTTP. We launch the SSL strip attack by running the modified caplet “hstshijack” in bettercap as seen in fig. 7

```

root@kali: ~
*
[82:25:28] [sys.log] [inf] hstshijack Reading SSL log ...
[82:25:28] [sys.log] [inf] hstshijack Reading caplet ...

Commands
hstshijack.show : Show module info.

Caplet
hstshijack.log > /usr/share/bettercap/caplets/hstshijack/xsl.log
hstshijack.ignore > *
hstshijack.targets > twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,www.linkedin.com
hstshijack.replacements > twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,www.linkedin.com

```

Figure 7: SSL Stripping

For demonstration, we used stackoverflow.com that can be seen to be a secured website as it has a padlock symbol at the topmost left as shown in fig. 8 before performing the stripping. After performing the stripping, one can see in fig. 9 that the padlock symbol is no longer there, and instead, “no secured” is written there because the Chrome browser usually warns users to be careful when using HTTP sites. The attacker can now read all information entered on this website indicating that the downgrade is successful as shown in fig. 10.

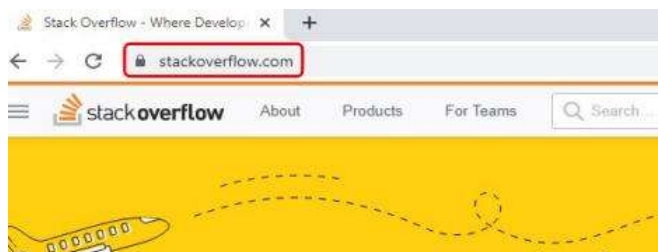


Figure 8: Website before Downgrading

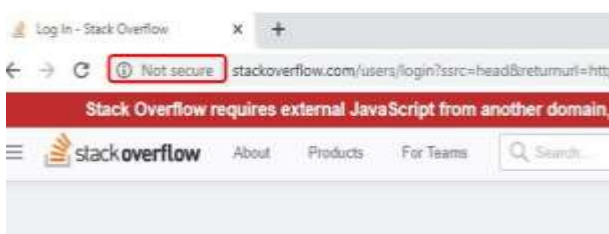


Figure 9: Website after Downgrading

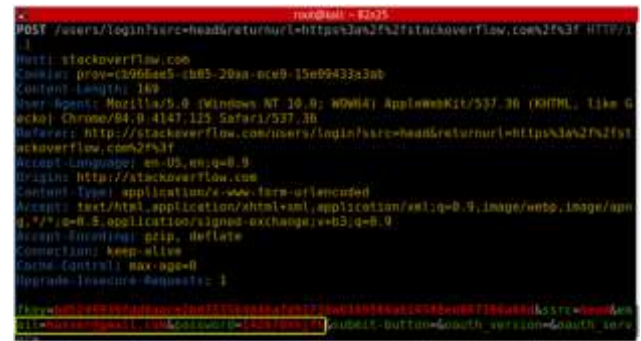


Figure 10: Captured Information after Downgrading

3.1. Detection Modules

Algorithm Detection in C Language: In the algorithm design, only ARP reply packets are captured and the headers used are seen below:

```

//ethernet header
struct ethhdr {
    unsigned char ethsrc[6];
    unsigned char ethdes[6];
    unsigned short ethhtype; };

// ARP header
struct ARPHdr {
    uint16_t arphardwaretype;
    uint16_t arpprototype;
    u_char arphardwaresize :6;
    u_char arpprotsz :4;
    uint16_t arpopcode;
    u_char arpMACsrc[6];
    u_char arpipsrc[4];
    u_char arpMACdes[6];
    u_char arpipdes[4]; };

```

Figure 11: Headers

The algorithm successfully captures fake packets with conflicting MAC addresses as shown in Fig. 12

In this figure, the gateway IP 172.17:0:1 is having two different MAC addresses (0C:7D:EB:BF:F4:00 and C2:01:0B:F6:00:01) and when we look at the ARP tables in Fig. 3 and fig. 4 we see that the MAC address of the gateway before the attack is C2:01:0B:F6:00:01 and after the attack is 0C:7D:EB:BF:F4:00. This tells us that someone is launching an attack and is pretending to be the gateway.

```

SRC IP: 172.17:00:01 DES IP:172.17:00:06 60
capture time is: Thu Aug 13 17:29:08 2020
SRCMAC address for ARP packet: 0C:7D:EB:BF:F4:00 DESMAC address for ARP pack

SRC IP: 172.17:00:01 DES IP:172.17:00:06 60
capture time is: Thu Aug 13 17:29:19 2020
SRCMAC address for ARP packet: C2:01:0B:F6:00:01 DESMAC address for ARP pack

```

Figure 12: Fake frames with conflicting MAC Addresses

3.2 Detection using Wireshark

Wireshark is a packet sniffer tool developed by Gerald Combs in 1988[17] to analyze and troubleshoot computer networks. In the expert information area of Wireshark, a warning is given that says “duplicate IP address configured (172.17.0.1)” as shown in fig 13. To further analyze this warning, we filter only ARP reply protocols with duplicate addresses with the Berkeley Packet Filter (BPF) syntax “arp.duplicate-address-detected and arp.opcode==2” which only filters ARP replies that have duplicate addresses. The details pane of the filtered packets gives us detailed information on what was seen in the expert information area. We can see that in frame 37268, it shows in fig 14 “Duplicate IP address detected for 172.17.0.1 (c2:01:0b:f6:00:01) also in use by 0c:7d:eb:bf:f4:00” in frame.

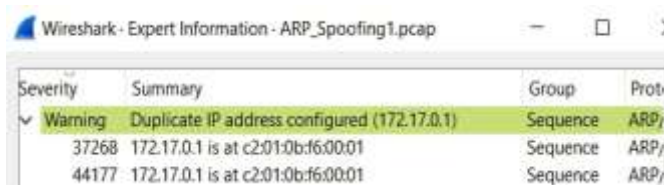


Figure 13 Duplicate Addresses

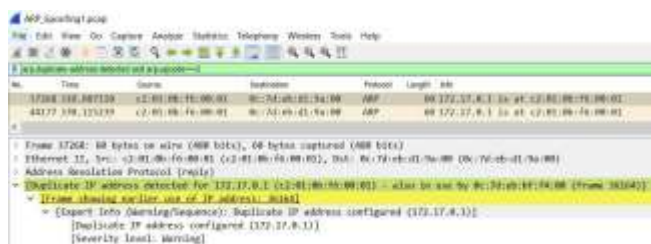


Figure 14. Details of Duplicate Addresses

4.0 PREVENTION

4.1 Prevention Module

In this module, the ARP spoofing attack is stopped by using the DHCP snooping and ARP inspection. DHCP snooping is a layer two feature that implements better security in an Ethernet switch environment by allowing its users to create both trusted and untrusted ports. Only trusted ports receive DHCP responses and are not monitored whereas untrusted ports are monitored for any suspicious activities and can't receive responses either [17][18][19]. It prevents rogues DHCP servers from creating fake DHCP offers to clients. The technique is a multiplayer technique in the sense that, not only is this technique used to prevent DHCP attacks but also, other security features such as Dynamic ARP Inspection (DAI) can make good use of this database to prevent ARP spoofing in the Ethernet environment. To use this technique, one has to enable it on the switch and once it is enabled, the switch builds a DHCP snooping binding database that includes: MAC addresses of hosts, leased IP addresses, leased time, binding time, and interface information associated with the hosts. Once the binding is done, ARP inspection intercepts all ARP

requests and all replies from all untrusted ports of DHCP snooping.

4.2 Algorithm for Prevention

Step 1: enable DHCP snooping

Step 2: configure all trusted ports in the topology (Gateway port Gi0/0)

Step 3: limit the transfer rate of frames from untrusted Ports (Gi0/2) to 10 frames per second

Step 4: enable Dynamic ARP Inspection in the particular LAN of interest

Step 5: set Gateway port as trust for successful communication between gateway and other hosts.

Step 6: if frames received from any untrusted port is more than 10 frames then

Drop frames and shut down port

else If IP address of untrusted port does not match with the IP address in the database then

Drop frames and shut down port

else if MAC address of untrusted port does not match the MAC address in the database then drop frames and and shut down port.

end if

Fig 15 shows that the port for the gateway Gigabit Ethernet (Gi0/0) has been configured to be trusted by DHCP snooping as stated in step two of the algorithm and the rate limit of untrusted ports which in this case is the attacker's port (Gi0/2) has been set to 10 as seen in fig. 16 this configuration is done to prevent the attacker from sending more than 10 frames per second thereby avoiding the attack.

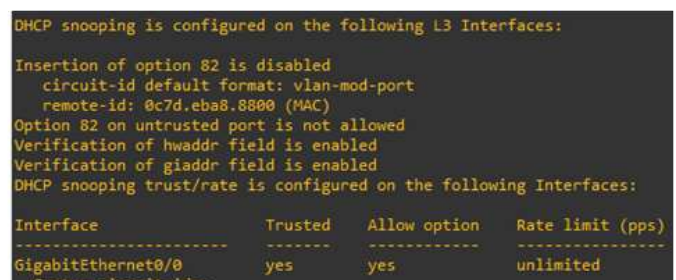


Figure 15 : Trusting Gateway Port

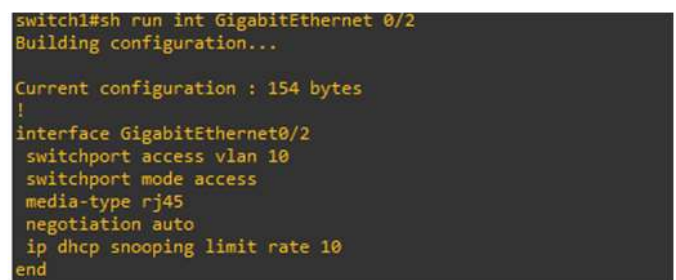


Figure 16: Untrusted Attacker's Port Limited

DAI is enabled and the gateway has been configured as a trusted port as stated in steps 4 and 5 and once enabled, frames are dropped from all other ports as shown in fig. 17 except the

gateway, which has been configured as trusted. The reason is that we are relying on DHCP snooping, whose database is empty as seen in fig. 18, at the moment, because no new IP addresses have been allocated since it was enabled. Due to this condition, the machines need to renew their <IP, MAC> through DHCP so that, the new addresses can now be added to the DHCP database as shown in fig. 19. This way, DAI can now use the database to work effectively.

```
Aug 13 05:35:21.327: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/2, vlan 10, (0c79)
Aug 13 05:35:22.329: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/2, vlan 10, (0c79)
Aug 13 05:35:23.073: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/2, vlan 10, (0c79)
Aug 13 05:35:45.798: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/2, vlan 10, (0c79)
Aug 13 05:35:46.889: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/2, vlan 10, (0c79)
```

Figure 17: DAI dropping all packets

```
Switch#sh ip dhcp snooping binding
MacAddress      IpAddress      Lease(sec)  Type           VLAN  Interface
-----
Total number of bindings: 0
```

Figure 18: No DHCP entry in database

After successfully building the DHCP snooping database, all hosts whose details are in the database are no longer denied gateway access and can connect to the gateway, as shown in fig. 20.

```
TC Thu Aug 13 2020]#sh ip dhcp snooping binding
MacAddress      IpAddress      Lease(sec)  Type           VLAN  Interface
-----
0C:7D:EB:BF:F4:00 172.17.0.5    85900      dhcp-snooping  10    GigabitEthernet0/2
0C:7D:FB:01:9A:00 172.17.0.6    85900      dhcp-snooping  10    GigabitEthernet0/1
08:5B:79:68:68:00 172.17.0.3    86381      dhcp-snooping  10    GigabitEthernet0/3
Total number of bindings: 3
```

Figure 19: filled DHCP database

Figure 20: Successful communication of all machines

4.3 ARP Spoofing Attack Failure

The attacker launches an ARP spoofing attack, and this time around, he is unsuccessful as seen in fig. 22 that ARP-inspection error is detected on the Gi0/2 that is the interface of the attacker machine. The data that the attacker is sending to the gateway and the victim machine to deceive them so that he can become the man in the middle has been dropped by the switch. Not only has it dropped the frames, but it has also shut down the port interface Gi0/2 that is being used by the attacker to perform the attack

```
Aug 13 10:46:27.076: %SW-DAL-4-FAKED-ARP-DEPR: 100 packets received in 27 milliseconds on Gi0/2.
Aug 13 10:46:28.061: %SW-DAL-4-FAKED-ARP-DEPR: 100 packets received in 27 milliseconds on Gi0/2, putting Gi0/2 in err-disabled state
Aug 13 10:46:28.129: %SW-DAL-4-FAKED-ARP-DEPR: 100 packets received in 27 milliseconds on Gi0/2, putting Gi0/2 in err-disabled state
Aug 13 10:46:28.437: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/1, vlan 10, (0c79)
Aug 13 10:46:28.437: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/1, vlan 10, (0c79)
Aug 13 10:46:28.437: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/1, vlan 10, (0c79)
Aug 13 10:46:28.437: %SW-DAL-4-DHCP-SNOOPING-DEPR: 1 Invalid ARP (Req) on Gi0/1, vlan 10, (0c79)
```

Figure 21: ARP Spoofing Prevented

5.0 CONCLUSIONS

This paper proposes a new approach for detecting and preventing ARP spoofing attacks in the LAN environment. In the proposed system, MITM attack over SSL is implemented using Bettercap and SSL strip, the attack is detected using Wireshark and Wincap algorithm in C language and has successfully been prevented using DHCP snooping and dynamic ARP inspection. This approach helps to ensure that enterprises run effectively and efficiently by creating reliability and ensuring security for network solidity.

REFERENCES

- [1] Yaibuates, Mayoan, and Rounsang Chaisricharoen. "A Combination of ICMP and ARP for DHCP Malicious Attack Identification." In *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pp. 15-19. IEEE, 2020.
- [2] Rupal, D. Raviya, DhavalSatsiya, Hresh Kumar, and Archit Agrawal. "Detection and prevention of ARP poisoning in dynamic IP Configuration." In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1240-1244. IEEE, 2016.
- [3] Jinhua, Gao, and Xia Kejian. "ARP spoofing detection algorithm using ICMP protocol." In *2013 International Conference on Computer Communication and Informatics*, pp. 1-6. IEEE, 2013.
- [4] Hao, Guo, and Guo Tao. "Principle of and Protection of Man-in-the-Middle Attack Based on ARP Spoofing." *JIPS* 5, no. 3 (2009): 131-134.
- [5] Duddu, Siromani, Ch LS Sowjanya, G. Ramakoteswara Rao, and Karthik Sainadh Siddabattula. "Secure Socket Layer Stripping Attack Using Address Resolution Protocol Spoofing." In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 973-978. IEEE, 2020.
- [6] Nayak, Gopinath, and Shefalika Ghosh Samaddar. "Different flavours of man-in-the-middle attack, consequences and feasible solutions." In *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5, pp. 491-495. IEEE, 2010.
- [7] Mahmood, Shahid, Syed Muhammad Mohsin, and Syed Muhammad AbrarAkber. "Network Security Issues of Data Link Layer: An Overview." In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1-6. IEEE, 2020.
- [8] Xing, Wenjian, Yunlan Zhao, and Tonglei Li. "Research on the defense Against ARP spoofing attacks based on Wincap." In *2010 Second International Workshop on Education Technology and Computer Science*, vol. 1, pp. 762-765. IEEE, 2010.
- [9] Ren, Ming, Yanhui Tian, Siqi Kong, Dali Zhou, and Danping Li. "An Detection algorithm for ARP man-in-the-middle attack based on data packet forwarding behavior characteristics." In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1599-1604. IEEE, 2020.
- [10] Hou, Xiangning, Zhiping Jiang, and Xinli Tian. "The Detection and Prevention for ARP Spoofing based on Snort." In *2010 International Conference on Computer Application and System Modeling (ICCSM 2010)*, vol. 5, pp. V5-137. IEEE, 2010.
- [11] M. Barnaba, "Anticap," 2003, Available at URL: <http://cvs.antifork.org/cvsweb.cgi/anticap>.
- [12] I. Teterin, "Antidote," 2002, Available at URL: <http://online.securityfocus.com/archive/1/299929>
- [13] Bruschi, Danilo, Alberto Ornaghi, and Emilia Rosti. "S-ARP: a secure address resolution protocol." In *19th Annual Computer Security Applications Conference, 2003. Proceedings.* pp. 66-74. IEEE, 2003.
- [14] WINPCAP: http://www.winpcap.org/docs/docs_412/html/main.html

- [15] Nagendran, K., Adithyan. A., Balaji.S., and Balakrishnan, S. "Sniffing HTTPS Traffic in LAN by Address Resolution Protocol Poisoning". International Journal of Pure and Applied Mathematics, ISSN No: 1314-3395. Volume 119 No 12 2018, 1187-1195.
- [16] Glăvan, Dragoș, Ciprian Răuciu, Radu Moinescu, and Sergiu Eftimie. "Man in the middle attack on HTTPS protocol." *Scientific Bulletin "Mircea cel Bătrân" Naval Academy* 23, no. 1 (2020): 199A-201.
- [17] Iqbal Haroon, and Sameena Naaz. "Wireshark as a Tool for Detection of Various LAN Attacks." *International Journal of Computer Sciences and Engineering*, Vol.7, Issue-5, pp. 833-837 2019.
- [18] Senecal, Louis. "Understanding and preventing attacks at layer 2 of the OSI reference model." In *4th Annual Communication Networks and Services Research Conference (CNSR'06)*, pp. 1-pp. IEEE, 2006
- [19] Nuhu, Abdulhafiz A., Faith O. Echobu, and Oyenike M. Olanrewaju. "Mitigating Dhcp Starvation Attack Using Snooping Technique." *Fudma Journal of Sciences* 4, no. 1 (2020): 560-566.



Gyarteng Emmanuel S. A who aims to achieve academic excellence was born in Ghana. He is currently a third year Computer Science undergraduate student at Southwest University of Science and Technology, China. His research interest is currently in the area of Artificial Intelligence. He is currently working on traffic light systems in reinforcement learning which presents the effective leverage in urban driving.



Hannah A. S. Adjei a graduate Member of IEEE since 2020 was born in Ghana West-Africa where she schooled and obtained a Bachelor of Science Degree in Telecommunications engineering in Ghana Technology University College in the year 2017. She studied Advanced Manufacturing and Informatization at Zhejiang University of Science and Technology Hangzhou, China from 2018 September to June 2019. She is currently studying Master of Engineering (Information

and Communications Engineering) at Southwest University of Science and Technology, China. In 2016, she worked with Enhanced Information Communication Centre, Techiman as an ICT infrastructure Systems Officer. After graduating in 2017, she worked as an intern for a year in Anointed Electricals Engineering Services, Accra as an Electrical Engineer. Her current research direction is in the area of Network Information Processing.



Associate Professor Tan Shunhua was born in China. He received his first degree in Computer Science and Technology Studies at Southwest University of Science and Technology in 1996 and later received his master's degree in software engineering from the University of Electronic Science and Technology of China in June 2008. He is mainly engaged in big data analysis and application technology research, network communication technology research, server and various terminal

equipment software developments. He has over the years participated in the completion of a number of national, provincial, and horizontal projects, and published more than 10 papers, including 2 papers in SCI journals and 4 papers in EI journals. At the same time, instructing students in scientific and technological activities to obtain a total of 16 national awards and 26 provincial awards.



George K. Agordzo likes seeking perfection through hard work and has a passion for technology and innovation. He received a 3-year Diploma in Education from the University of Cape Coast, Ghana in the year 2011. In 2017, he graduated with Bachelor's Degree in Information Technology from Valley View University - Oyibi, Ghana. He is currently a Graduate Student of Master of Engineering (M.Eng) in Advanced Manufacturing and Informatization, a Computer Science and

Engineering based program at the Zhejiang University of Science and Technology, Hangzhou-China. From 2011 to 2015, he served as an IT Instructor in Ghana. Until June 2018, he was the System Administrator for the USAID Learning Team and EMIS, GES in the Ho Municipality. He loves to explore new opportunities and also a Member of IEEE. His field of research has been identified with Software Engineering, Data mining, Machine Learning, Data Analysis, Artificial Intelligence, Computer Vision, Image processing, and classification. <https://orcid.org/0000-0002-6818-3718>