2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM) | 979-8-3503-1600-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/NCIM59001.2023.10212773

An Effective Method to Detect DHCP Starvation Attack using Port Scanning

Asaduzzaman Jony[1], Abu Saleh Musa Miah[2], and Muhammad Nazrul Islam[3]
[1]*Dept. of Computer Science and Engineering, Bangladesh University of Professionals (BUP)*, Dhaka, Bangladesh
[2]*Dept. of Computer Science and Engineering, The University of Aizu*, Aizuwakamatsu, Japan
[3]*Dept. of Computer Science and Engineering, Military Institute of Science and Technology (MIST)*, Dhaka, Bangladesh
Email: nazrul@cse.mist.ac.bd

***Abstract*—Nowadays, the DHCP starvation attack is a crucial attack that denied the services for the legitimate network user where the attacker mainly occupies all the IP addresses in the DHCP server pool. Most of the existing studies proposed ICMP (Internet Control Message Protocol) and the ARP (Address Resolution Protocol) based techniques to detect the DHCP starvation attack. However, the ICMP-based detection is ineffective if the client has an active host-based firewall that blocks the ICMP ECHO REQUEST, while the ARP-based detection does not work for DHCP networks separated by a DHCP relay agent and works only within the same or local network. Therefore, the objective of this research is to propose an effective technique to detect DHCP starvation attack detection through port scanning to address the limitations of ICMP and ARP-based detection methods. To attain this objective, this study first highlights the limitations of existing detection methods demonstrating insightful facts. Secondly, the study proposed a port scan-based DHCP starvation attack detection method for local and remote (relay) DHCP networks. Finally, the effectiveness and validation of the proposed techniques were demonstrated using the Multivendor Network Emulation Software (EVE-NG). The study showed that the proposed port scanning-based technique could accurately detect the DHCP starvation attack while overcoming the limitations of the ICMP and ARP techniques.**

***Index Terms*—DHCP starvation, Attack simulation, Attack signature, Port Scan, and Network Security.**

I. Introduction

Security become key concern for every IT innovation [1], [2], while a number of security issues exist related to DHCP (Dynamic Host Configuration Protocol) operation. An IP address is generally assigned to a device statically or dynamically, while assigning the IP addresses dynamically is more convenient and reduces administrative effort and the chances of misconfigurations. This is where the DHCP emerges. Generally, a DHCP client exchanges four messages (DISCOVER, OFFER, REQUEST and ACKNOWLEDGEMENT) with the DHCP server to obtain an IP address from the DHCP server [3], [4]. An attacker connects to the LAN and sends many DHCPDISCOVER messages with forged MAC addresses. The DHCP server receives each DHCPDISCOVER message and responds to all forged MAC addresses pretending to be legitimate clients. After some time, the DHCP server pool runs out of IP addresses and fails to provide an IP address to a legitimate client. Hence, it is known as a DHCP starvation attack, a denial-of-service (DoS) attack [5], [6].

An ICMP (Internet Control Message Protocol, RFC 792) based DHCP starvation attack detection has been proposed by Mayoon and Rounngsan [7]. But their proposed method is ineffective if the destination host has an active firewall blocking the ICMP ECHO REQUEST. To overcome this issue, Yaibuates and Chaisricharoen [8] have proposed a detection method using a combination of ICMP and ARP (Address Resolution Protocol, RFC 6747). Though ARP-based detection partially solves this issue but fails to detect legitimate clients in different networks separated by a DHCP relay agent. On the other hand, Nikhil and Neminath [4] proposed a machine learning-based DHCP starvation detection method. Their proposed method first generates a profile of regular DHCP operations. Then, the method compares the current DHCP operation with the profile to detect a DHCP starvation attack. However, DHCP starvation attacks can be carried out using different tools [9]. Moreover, it is also possible to broadcast DHCPDISCOVER packets after a certain time interval in order to look like a normal DHCP operation. Hence, Nikhil and Neminath's proposed method is ineffective in detecting slow or stealth DHCP starvation attacks.

Therefore, this study aims to propose an effective DHCP starvation attack detection method using a port scanning technique to address the limitations of ICMP and ARP-based detection methods. The proposed port scan technique scans popular 1000 TCP ports against each IP address assigned by the DHCP server [10]. The IP addresses with at least one open port are determined as legitimate DHCP clients. Next, the background and realted studies are discussed followed by the proposed DHCP starvation attack detection method and its evaluation are discussed. In Conclusions, the limitations, future work and concluding remarks are briefly presented.

II. RELATED STUDIES

A. ICMP-based detection and it's limitations

A method to detect the DHCP starvation attack using ICMP is proposed in [7]. Their proposed detection method initiates ICMP ECHO REQUEST to all the consumed IP addresses in the DHCP POOL. If a particular IP address responds with ICMP ECHO REPLY, then it is determined that a legitimate host has consumed this particular IP address. However, if any IP address does not respond with ICMP ECHO REPLY, it is determined that it is occupied through a DHCP starvation

979-8-3503-1600-1/23/$31.00 ©2023 IEEE

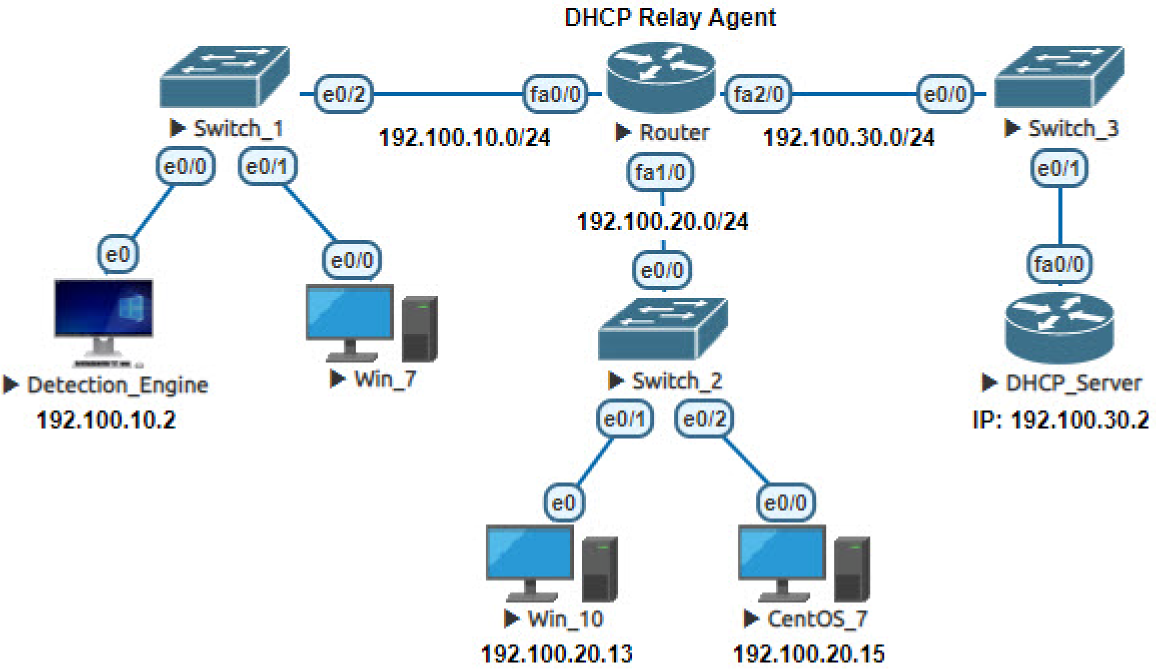


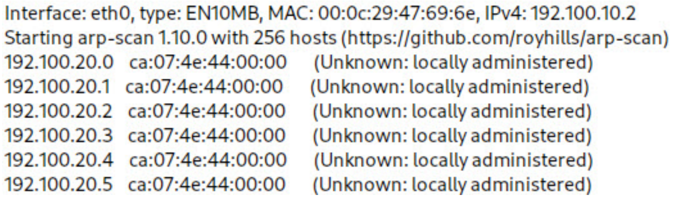
Fig. 1. Multiple DHCP networks separated by DHCP relay agent

attack. However, some operating system has a native firewall that blocks ICMP ECHO REQUEST. For instance, the Windows operating system has a built-in host-based firewall called ”windows defender”. By default, windows defender denies the PING or ICMP ECHO REQUEST. In this scenario, the Windows host does not send ICMP ECHO REPLY, even if the host is live. Hence, using ICMP to determine the legitimacy of any host is inaccurate when the destination firewall is blocking ICMP.

B. *ARP-based detection and itś limitations*

An approach that combines the ICMP and ARP (Address Resolution Protocol) to detecting the DHCP starvation attack is proposed in [8]. Here, if any client declines ICMP or PING, then it will initiate an ARP scan. As ARP packets are required to obtain the destination device’s MAC address, ARP packets are allowed through firewalls. Even if any DHCP client declines ICMP but replies to ARP REQUEST, then itś considered a legitimate client. Failing to reply to both ICMP and ARP indicates that a DHCP starvation attack occupies the IP address. According to Fig. 1, the DHCP server resides in a different network, and 192.100.10.0/24 and 192.100.20.0/24 DHCP networks are connected with the DHCP relay agent [11]. Hence, the relay agent or router has created three different broadcast domains. Routers do not allow any broadcast packet from one broadcast domain to another. However, the arp-scan tool is used to broadcast ARP REQUESTs to 192.100.20.0/24 network from a detection engine (192.100.10.2) in 192.100.10.0/24 network through interface eth0. ARP REQUESTs broadcast from 192.100.10.2 in the 192.100.10.0/24 network have failed to reach 192.100.20.0/24 (Fig. 2).

```
# sudo arp-scan -I eth0 192.100.20.0/24
```



```
Interface: eth0, type: EN10MB, MAC: 00:0c:29:47:69:6e, IPv4: 192.100.10.2
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.100.20.0   ca:07:4e:44:00:00    (Unknown: locally administered)
192.100.20.1   ca:07:4e:44:00:00    (Unknown: locally administered)
192.100.20.2   ca:07:4e:44:00:00    (Unknown: locally administered)
192.100.20.3   ca:07:4e:44:00:00    (Unknown: locally administered)
192.100.20.4   ca:07:4e:44:00:00    (Unknown: locally administered)
192.100.20.5   ca:07:4e:44:00:00    (Unknown: locally administered)
```

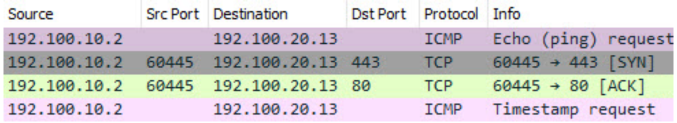
Fig. 2. the DHCP relay agent declines ARP broadcasts for 192.100.20.0/24 network

According to Figure 2, the DHCP relay agent has received ARP broadcasts on fa 0/0 interface and drops all ARP REQUESTS. In this scenario, the legitimacy of any DHCP client residing in the 192.100.20.0/24 network that is blocking ICMP or PING cannot be determined using this detection method. Even if that particular host is live, its IP address is detected as occupied by a DHCP starvation attack. Thus their proposed detection method causes false detection of DHCP starvation attacks in large enterprise networks where a single DHCP server provides an IP address to several DHCP networks segregated by a DHCP relay agent.

C. *Port Scan operation*

Ports are software-based virtual points where network connections start and end. They are managed by the computer’s operating system and represent individual services running on that machine. According to IANA, there is a total of 65,535 ports; among them, ports 0 to port 1023 are called system ports [12]. Port Scan is a technique to scan the ports of an individual or multiple hosts in any network. Port scan works by sending packets to a particular port or ports and examining the reply [13]. Depending on the port's status, the scan mainly determines a port as open, closed, or filtered. An open port means an application is actively accepting TCP connections, UDP datagrams, or SCTP associations on that port. Sending an SYN packet to any open port gives an SYN-ACK packet as a reply. A closed port means the port is accessible, but no application is listening on that port. Sending an SYN packet to any closed port gives RST a reply. However, a filtered port means that the scan cannot determine whether the port is open or closed, as the firewall prevents the scan probes from reaching the port.

In this study, we have used a port scanning application known as "Nmap" (Network Mapper). A basic Nmap scan (nmap destination IP) initiated by the root user sends four packets to the destination. They are (a) ICMP echo request, (b) SYN packet to TCP port 443, (c) ACK packet to TCP port 80, and (d) ICMP timestamp request (Fig. 3).



Source	Src Port	Destination	Dst Port	Protocol	Info
192.100.10.2		192.100.20.13		ICMP	Echo (ping) request
192.100.10.2	60445	192.100.20.13	443	TCP	60445 → 443 [SYN]
192.100.10.2	60445	192.100.20.13	80	TCP	60445 → 80 [ACK]
192.100.10.2		192.100.20.13		ICMP	Timestamp request

Fig. 3. Four packets sent to the destination host by basic nmap scan

If the destination replies to any of the ICMP messages or any of TCP ports 80 or 443 is open, then nmap determines the destination host as live. However, in most cases, the destination host's firewall blocks the ICMP, and TCP ports 80 and 443 are closed unless the host is running a web server on it. Hence, even if the host is actually live, the basic nmap scan is insufficient to determine whether the host is live or not. In this case, an advanced nmap scan is required. To overcome this limitation, we have used the nmap host discovery technique with "no ping" option (nmap –Pn destination IP) to determine whether the host is live. While using this option, the scanning host does not send any ICMP or PING to the destination host. Instead, it sends SYN packets to popular 1000 TCP ports on the destination host to discover open TCP ports. It performs a "TCP half-open scan" for any of these 1000 TCP ports that replies SYN, ACK to the initial SYN packet.

According to the TCP 3-way handshake [14], the source sends a TCP SYN packet to the destination. The SYN packet is sent to a particular TCP port or range of ports. If the port is OPEN, the destination sends back an SYN-ACK packet and expects an ACK packet from the source. However, as this option (-Pn) initiates a "TCP half-open scan", so source does not complete the 3rd step of the TCP handshake. Instead of sending an ACK packet, the source sends RST (Reset) packet forcefully to terminate the TCP connection (Figure 4). This scan aims to get the SYN, ACK packets from the open TCP ports without establishing a TCP session with them.

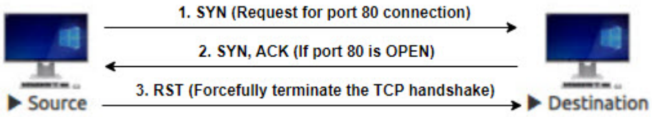

1. SYN (Request for port 80 connection)
2. SYN, ACK (If port 80 is OPEN)
3. RST (Forcefully terminate the TCP handshake)
▶ Source
▶ Destination

Fig. 4. TCP half-open scan for port 80

III. DHCP STARVATION ATTACK & SIGNATURE

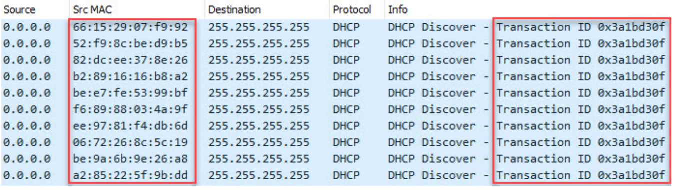
DHCP (RFC 2131) has no authentication system to verify the legitimacy of DHCP clients. This weakness makes the DHCP server very susceptible to DHCP starvation attack. At first, the attacker connects his device with the target network, where he/she wants to exhaust IP addresses from the DHCP server pool. Once connected, the attacker broadcasts many DHCP DISCOVER packets into the network. The DHCP server receives all the DHCPDISCOVOER packets and assigns the IP address against each DHCPDISCOVOER packet it receives until the DHCP server IP address pool is completely occupied. As the IP address pool is occupied by forged clients crafted by attackers, hence, authentic clients fail to obtain an IP address from the DHCP server. This situation is known as "DHCP starvation". DHCP starvation attacks could be carried out using a variety of tools in various environments, like wired, wireless, SDN and virtualization environments [**?**], [15], [16]. In this study, using a unique tool, the DHCP starvation attack has been demonstrated practically in EVE-NG virtual lab using KALI Linux. Wireshark, a packet analyzer, is used to analyze the signature of a DHCP starvation attack. The KALI Linux has only one attacking interface with a MAC address of 00:0c:29:47:69:6e.

"Amateras" is an updated tool that offers customized DHCP starvation attacks. This tool only works in CLI mode and does not require root privilege to execute. The attacker can customize the command like, in which interface he/she is initiating the attack and the range of IP address (192.100.10.1 to 192.100.10.254) he/she wants to occupy from the DHCP server pool. The following command initiates the attack-

```
./Amateras run -iface eth0 -start 192.100.10.1 -end
192.100.10.254 –verbose
```

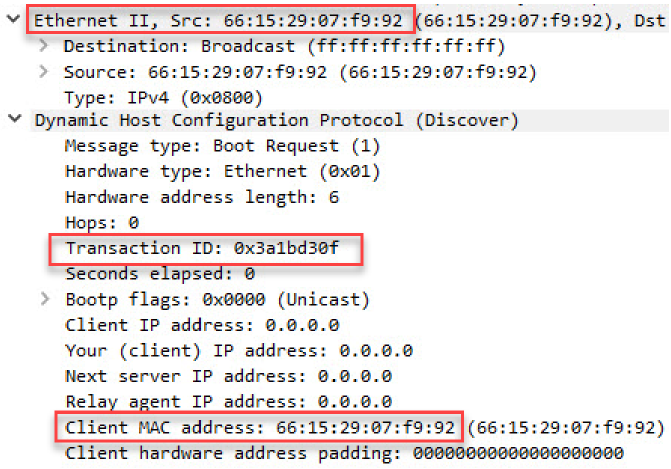
Once the attack is launched, the DHCP server is flooded with tons of DHCPDISCOVER messages (See figure 5). This figure showed that "Amateras" assigns identical transaction IDs (xid) for all DHCPDISCOVER packets. While analyzing an individual DHCPDISCOVER packet generated by "Amateras" (See Figure 6), it is found that the source MAC address in the Ethernet frame header is different from the actual MAC address (00:0c:29:47:69:6e) of the attacking interface. In addition, this tool puts a random source MAC address as the client MAC address (chaddr).

Different DHCP starvation tools have different attack signatures. In addition, some DHCP starvation tools are able to broadcast DHCPDISCOVER messages after certain time interval to avoid traditional detection methods that works



```
Source    Src MAC            Destination      Protocol  Info
0.0.0.0   66:15:29:07:f9:92  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   52:f9:8c:be:d9:b5  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   82:dc:ee:37:8e:26  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   b2:89:16:16:b8:a2  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   be:e7:fe:53:99:bf  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   f6:89:88:03:4a:9f  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   ee:97:81:f4:db:6d  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   06:72:26:8c:5c:19  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   be:9a:6b:9e:26:a8  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
0.0.0.0   a2:85:22:5f:9b:dd  255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x3a1bd30f
```

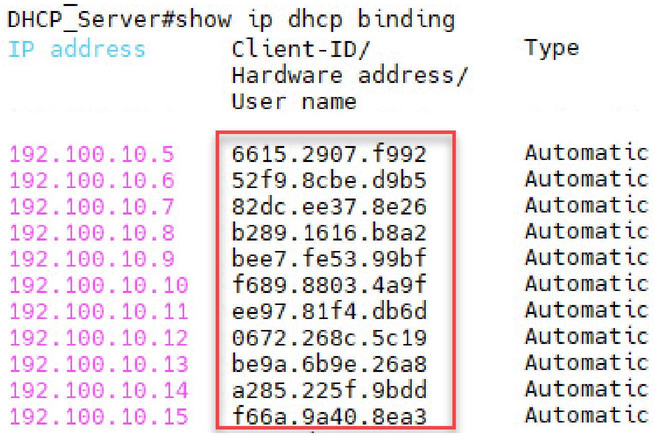
Fig. 5. Amateras broadcasts DHCPDISCOVER messages with random source MAC and the identical transaction is (xid)



```
v Ethernet II, Src: 66:15:29:07:f9:92 (66:15:29:07:f9:92), Dst
   > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
   > Source: 66:15:29:07:f9:92 (66:15:29:07:f9:92)
     Type: IPv4 (0x0800)
v Dynamic Host Configuration Protocol (Discover)
     Message type: Boot Request (1)
     Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
     Transaction ID: 0x3a1bd30f
     Seconds elapsed: 0
   > Bootp flags: 0x0000 (Unicast)
     Client IP address: 0.0.0.0
     Your (client) IP address: 0.0.0.0
     Next server IP address: 0.0.0.0
     Relay agent IP address: 0.0.0.0
     Client MAC address: 66:15:29:07:f9:92 (66:15:29:07:f9:92)
     Client hardware address padding: 00000000000000000000
```

Fig. 6. DHCP starvation attack signature analysis (Amateras)

on counting DHCPDISCOVER messages exchanged within a specific time. Hence, it is very difficult to detect DHCP starvation attack based on attack signature only. Figure 7 shows that, IP address of CISCO based DHCP server are occupied with forged clients just after few seconds of DHCP starvation attack.



```
DHCP_Server#show ip dhcp binding
IP address         Client-ID/              Type
                   Hardware address/
                   User name

192.100.10.5       6615.2907.f992          Automatic
192.100.10.6       52f9.8cbe.d9b5          Automatic
192.100.10.7       82dc.ee37.8e26          Automatic
192.100.10.8       b289.1616.b8a2          Automatic
192.100.10.9       bee7.fe53.99bf          Automatic
192.100.10.10      f689.8803.4a9f          Automatic
192.100.10.11      ee97.81f4.db6d          Automatic
192.100.10.12      0672.268c.5c19          Automatic
192.100.10.13      be9a.6b9e.26a8          Automatic
192.100.10.14      a285.225f.9bdd          Automatic
192.100.10.15      f66a.9a40.8ea3          Automatic
```

Fig. 7. IP address pool of DHCP server occupied with fake hosts

IV. PROPOSED DETECTION METHOD

In this research, an effective method is proposed to detect the DHCP starvation attack that overcomes the limitations of ICMP and ARP based detection methods. The proposed method is capable to detect DHCP starvation attack in local as well as remote networks. The procedure of the proposed method is presented below: Firstly, the clients in DHCP network exchanges DISCOVER, OFFER, REQUEST and ACK messages with the DHCP server in order to successfully obtain an IP address from the DHCP server.

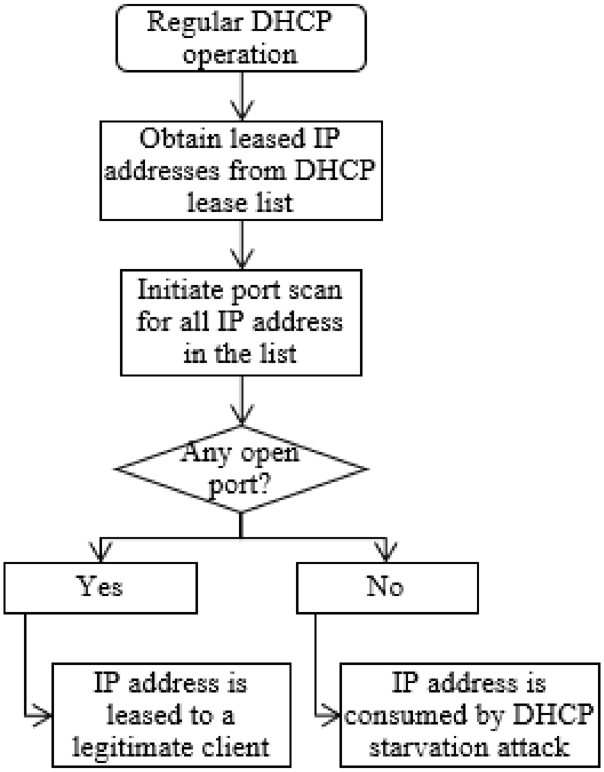

Regular DHCP operation
Obtain leased IP addresses from DHCP lease list
Initiate port scan for all IP address in the list
Any open port?
Yes
No
IP address is leased to a legitimate client
IP address is consumed by DHCP starvation attack

Fig. 8. Flowchart of the proposed (port scan based) detection method

Secondly, the DHCP server assigns IP addresses to clients for a specific time which is known as DHCP lease time. The DHCP server maintains a database called "DHCP lease list" where it stores the provided IP address, client's name /MAC address and lease expiration time. Hence, the list of IP addresses assigned by DHCP server is achieved from the DHCP lease list.

Thirdly, once the list of assigned IP address is found, the detection engine initiates proposed port scan for all of the IP addresses in the list. The detection engine sends SYN packet to popular 1000 TCP ports for each IP addresses stored in the list. Depending on the response of SYN packet, the detection engine determines whether an IP address is assigned to a legitimate host or occupied by DHCP starvation attack.

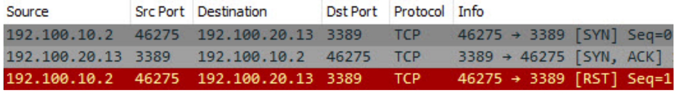
Finally, once the detection engine receives a SYN, ACK reply from any of the 1000 TCP ports of a particular IP address, it determines that IP address is assigned to a legitimate host. Because, based on the operating system, every host has some open ports, while some are kept open for the system services and some are open for specific applications. In case, the detection engine does not receive any SYN, ACK from any of 1000 TCP ports of an IP address, then it identifies the IP address is consumed by DHCP starvation attack.

V. EVALUATING THE PROPOSED METHOD

To evaluate the performance of the proposed detection method the EVE-NG was used, while the port scan was initiated from a Microsoft windows 10 based detection engine. Windows 10, and CentOS 7 based systems were used as a DHCP clients, and the CISCO IOS 7200 series router was used as a DHCP server and a DHCP relay agent. From the DHCP server IP lease list, it is found that 2 hosts (192.100.20.13 and 192.100.20.15) have obtained IP addresses through DHCP. From the detection engine (192.100.10.2) port scan was initiated for 192.100.20.13 (Windows 10) and 192.100.20.15 (CentOS 7). Again, the following port scan command with the –Pn option initiates TCP half-open scan for the 1000 TCP ports on 192.100.20.13 and 192.100.20.15# nmap -Pn 192.100.20.13 192.100.20.15

A. *Checking windows 10 host (192.100.20.13)*

The detection engine (192.100.10.2) initiates port scan and sends SYN packets to the popular 1000 TCP ports of both IP addresses. Windows 10 host (192.100.10.13) has Microsoft RDP (Remote Desktop Protocol) service open, which was running on TCP port 3389. The proposed nmap scan randomly sends SYN packets to popular 1000 TCP ports of 192.100.20.13. Wireshark trace report of 192.100.20.13 is shown in Figure 9.



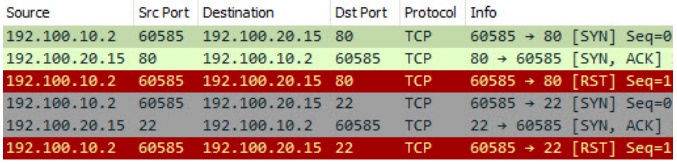
Source	Src Port	Destination	Dst Port	Protocol	Info
192.100.10.2	46275	192.100.20.13	3389	TCP	46275 → 3389 [SYN] Seq=0
192.100.20.13	3389	192.100.10.2	46275	TCP	3389 → 46275 [SYN, ACK]
192.100.10.2	46275	192.100.20.13	3389	TCP	46275 → 3389 [RST] Seq=1

Fig. 9. Detection of 192.100.20.13 as a legitimate DHCP client

1) The detection engine or source (192.100.10.2) sends a SYN packet to the destination (Windows 10, 192.100.20.13) on TCP port 3389 using a random source port 46275.
2) As TCP port 3389 is open on the destination host, it sends a SYN, ACK packet reply to the source. This reply comes from TCP port 3389 to TCP port 46275. The remaining 999 filtered TCP ports just dropped the SYN packet without sending any reply.
3) As the source initiated TCP half-open scan (see Figure 4), it sends a RST (Reset) packet to the TCP 3389 port of the destination device using the same source port 46275. It forcefully terminates the TCP handshake.

B. *Checking CentOS 7 host (192.100.20.15)*

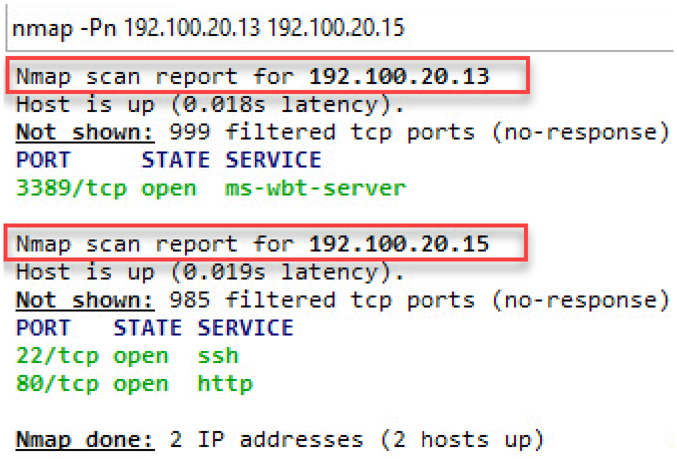
Similarly, the detection engine again sends SYN packets to CentOS 7 host (192.100.20.15). As 192.200.20.15 has two open TCP ports (80, 22) hence, both TCP ports reply SYN, ACK as a response of the SYN packet. However, the source terminates the TCP handshake by sending RST (Reset) to both TCP ports.



Source	Src Port	Destination	Dst Port	Protocol	Info
192.100.10.2	60585	192.100.20.15	80	TCP	60585 → 80 [SYN] Seq=0
192.100.20.15	80	192.100.10.2	60585	TCP	80 → 60585 [SYN, ACK]
192.100.10.2	60585	192.100.20.15	80	TCP	60585 → 80 [RST] Seq=1
192.100.10.2	60585	192.100.20.15	22	TCP	60585 → 22 [SYN] Seq=0
192.100.20.15	22	192.100.10.2	60585	TCP	22 → 60585 [SYN, ACK]
192.100.10.2	60585	192.100.20.15	22	TCP	60585 → 22 [RST] Seq=1

Fig. 10. Detection of 192.100.20.15 as a legitimate DHCP client

According to Figure 10, the detection engine has found SYN, and ACK responses from TCP ports 80 and 22 from the destination (192.100.20.15). Hence, it determines that the destination (192.100.20.15) is also a legitimate DHCP client. The output of the port scan is shown in Figure 11. Thus the detection engine initiates a port scan for all the IP addresses in the DHCP server IP lease list and evaluates the result. Again, the accuracy of ICMP, a combination of ICMP & ARP and the proposed port scan-based detection are explored. The DHCP starvation attack detection efficiency was tested in a live enterprise network where 2 DHCP classless private networks are segregated by a DHCP relay agent. There are total of 30 usable IP addresses in each network, and all the scans are performed from a host connected to Network 1 (172.10.10.0 /27). Detection error rate is measured following the equation (1).



```
nmap -Pn 192.100.20.13 192.100.20.15

Nmap scan report for 192.100.20.13
Host is up (0.018s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server

Nmap scan report for 192.100.20.15
Host is up (0.019s latency).
Not shown: 985 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http

Nmap done: 2 IP addresses (2 hosts up)
```

Fig. 11. Port scan determines 192.100.20.13 & 192.100.20.15 as legitimate hosts

In case of Network 1 (172.10.10.0 /27), the DHCP lease list shows that all the IP addresses ranging from 172.10.10.1-172.10.10.30 (30 IP addresses) are occupied by DHCP clients, where the number of actual live hosts and starvation victim are 16 and 14, respectively. Similarly, for Network 2 (192.10.10.0 /27), the DHCP lease list shows all the IP addresses ranging from 192.10.10.1 to 192.10.10.30 (30 IP addresses) are occupied by DHCP clients, where the number of actual live hosts and starvation victim are 12 and 18, respectively.

$$\phi = \frac{\text{Detected as victims} - \text{Actual victims}}{\text{Actual victims}} \times 100 \qquad (1)$$

TABLE I
DETECTION ERROR RATE IN LOCAL NETWORK 172.10.10.0 /27.

Subjects	**ICMP**	**ICMP & ARP**	**PORT SCAN**
Detected as Live	10	16	16
Detected as victim (Total – detected as live)	20	14	14
Detection Error Rate, ϕ	42.8%	0.0%	0.0%

TABLE II
DETECTION ERROR RATE TESTED IN REMOTE NETWORK 192.10.10.0 /27.

Subjects	**ICMP**	**ICMP & ARP**	**PORT SCAN**
Detected as Live	07	07	11
Detected as victim (Total – detected as live)	23	23	19
Detection Error Rate, ϕ	27.7%	27.7%	5.5%

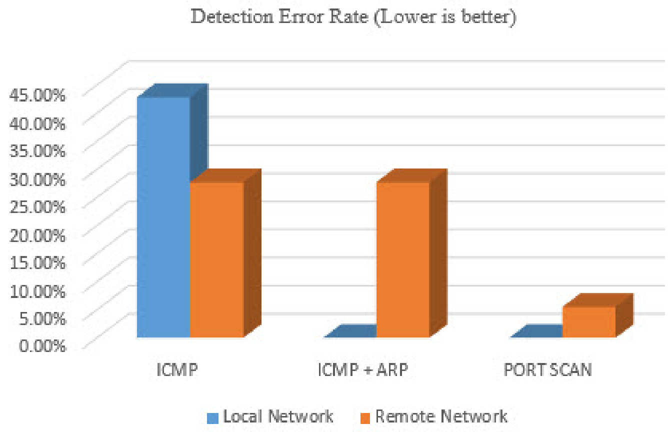


Fig. 12. Detection error rate comparison of all detection methods

The remote network was segregated by a DHCP relay agent that denies ARP broadcasts. Hence, the scan result for the remote network shows the same detection error rate for both ICMP, and ICMP and ARP-based detection. Here, the ICMP-based detection shows significant error while determining the victim IP addresses of the DHCP starvation attack (Figure 12). On the other hand, the combination of ICMP and ARP-based detection performs well in the local network. Still, it performs poorly while scanning for IP addresses occupied by DHCP starvation attacks in the remote network. However, the proposed port scan-based detection method provides the lowest error while detecting DHCP starvation attacks in local and remote DHCP networks. The proposed port scan is a manual process. There is no defined scan interval. However, the traffic generated by this scan depends on the number of IP address the user is scanning for.

VI. CONCLUSION

The DHCP is an insecure protocol susceptible to DoS attacks like DHCP starvation. This study thus demonstrated a DHCP starvation attack and its signatures. The existing ICMP-based detection is not fruitful when the destination host has a firewall that blocks ICMP; while the ARP-based detection is only effective for the local DHCP network. Thus, these are ineffective for determining legitimate DHCP clients segregated by a DHCP relay agent. The proposed port scan-based detection method showed very effectiveness for both the local and remote DHCP networks for determining DHCP starvation attack. However, any particular host that is blocking all the TCP ports can't be determined using the proposed method. In addition, scanning 1000 TCP ports against each IP address takes significant amount time. In the future, a python based DHCP starvation detection engine could be developed that can craft SYN packets using "scapy" and able to scan more ports faster than nmap. Hence, it will enhance the efficiency of detecting DHCP starvation attack.

REFERENCES

[1] M. N. Islam, M. M. H. Mia, M. F. I. Chowdhury and M. A. Matin, "Effect of Security Increment to Symmetric Data Encryption through AES Methodology," *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Phuket, Thailand, 2008, pp. 291-294

[2] T. Zaki, M. S. Uddin, M. M. Hasan and M. N. Islam, "Security threats for big data: A study on Enron e-mail dataset," *International Conference on Research and Innovation in Information Systems*, Langkawi, 2017.

[3] R. Droms, "Automated configuration of TCP/IP with DHCP," *IEEE Internet Computing*, vol. 3, no. 4, pp. 45 - 53, 1999.

[4] N. Tripathi and N. Hubballi, "A probabilistic anomaly detection scheme to detect DHCP starvation attacks," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bangalore, India, 2016.

[5] P. Anu and S. Vimala, "A survey on sniffing attacks on computer networks," in *2017 International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, India, 2017.

[6] A. Sadiqui, *Computer Network Security*. London, England: ISTE Ltd and John Wiley Sons, 2020.

[7] M. Yaibuates and R. Chaisricharoen, "ICMP based Malicious Attack Identification Method for DHCP," in *The 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)*, Chiang Rai, Thailand, 2014.

[8] M. Yaibuates and R. 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*, Pattaya, Thailand, 2020.

[9] M. Aldaoud, D. Al-Abri, A. Al Maashri, and F. Kausar, "DHCP attacking tools: an analysis," *Journal of Computer Virology and Hacking Techniques*, p. 119–129, 2021.

[10] K. Hess, "Nmap command line info gathering magic," 4 March 2020. [Online]. Available at: https://www.redhat.com/sysadmin/nmap-info.

[11] W. A. Syafei, Y. A. Adi Soetrisno and A. B. Prasetijo, "Centralized Dynamic Host Configuration Protocol and Relay Agent for Smart Wireless Router," *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, Semarang, Indonesia, pp. 1-5, 2019.

[12] IANA, "Service Name and Transport Protocol Port Number Registry," [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[13] M. Asaduzzaman, P. P. Rawshan, N. N. Liya, M. N. Islam, and N. K. Dutta, "A vulnerability detection framework for CMS using port scanning technique," in *Cyber Security and Computer Science*, Springer, 2020, pp. 128–139.

[14] R. Hunt, "Transmission Control Protocol/Internet Protocol (TCP/IP)," in *Encyclopedia of Information Systems*, Elsevier, 2003, pp. 489-510.

[15] C. Toprak, C. Turker and A. T. Erman, "Detection of DHCP Starvation Attacks in Software Defined Networks: A Case Study," *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, Bosnia and Herzegovina, pp. 636-641, 2018.

[16] J. -L. Wang and Y. -C. Chen, "An SDN-based defensive solution against DHCP attacks in the virtualization environment," *2017 IEEE Conference on Dependable and Secure Computing*, Taipei, Taiwan, pp. 529-530, 2017.