Contents

What Happened :

Essential Information:

Looking at the Expert Information in Wireshark the capture reveals that there are multiple MAC addresses being used by the IP address 10.23.0.151, as indicated by the "Duplicate IP address (10.23.0.151)" warning. At packet number 6, a user joins the multicast group 224.0.0.251 and requests to receive multicast traffic with MAC address f0:49:41:83:63:8c. In order to communicate, the user needs to be assigned an IP address. Packets 7 and 8 show that zyxelcom_41:80:dc sends a request for the MAC address associated with IP address 10.23.0.151. When it doesn't receive a response, zyxelcom_41:80:dc offers the IP address 10.23.0.151 to the MAC address f0:49:41:83:63:8c. The IP address is assigned to the user's MAC address, as confirmed in packet 10. This indicates that the router, zyxelcom_41:80:dc (IP address 10.23.0.1), has assigned the IP address 10.23.0.151 to the MAC address f0:49:41:83:63:8c. Name Michael (Potential exploiter/attacker).

Analysing further there seems to be a DHCP exhaustion/starvation, this is notable in the DHCP ACK packets(44,52 etc), where the host's name is Michael throughout in the application layer. In DHCP exhaustion attacks, the DHCP server's pool of available IP addresses is depleted, causing legitimate client devices to be unable to obtain IP addresses and connect to the network(packet 606 router unable to assign IP address to victim). This can create an opportunity for attackers to carry out ARP spoofing attacks or a rogue server attack. This is evidence by router assigning Ip address from packet 8 throughout to 606 after this Ip addresses were not assigned as the DHCP server was Exhausted. Which allowed the attacker to setup their rogue server tricking the victim(ee:fd:34:6b:72:1c). This was all set up by the attacker. It is also evidence that router did not acknowledge an Ip address at packet 606 meaning it was out of Ip addresses. It is clear that Michael is involved in the attack as Michael is the person exhaustion the DHCP server through series of DHCP requests. Michael joins the network first then performs the attack, the DHCP starvation in total took around 8 minutes to perform without Network team noticing.

Recording 1:

A new user with the MAC address ee:fd:34:6b:72:1c has joined the network and sent a broadcast requesting an IP address using a DHCP discovery packet (packet 598). However, packet 599 from f0:2c:0f:b2:08:52 shows suspicious behaviour, as it appears to be searching for any DHCP discoveries, and this occurred right after ee:fd:34:6b:72:1c's request. This may indicate that f0:2c:0f:b2:08:52 is attempting to act as a rogue DHCP server and is trying to view DHCP packets. In packet 603, an unorthodox DHCP offer is made by the IP address 10.23.0.151, which raised suspicions since only the legitimate DHCP server (router) should assign IP addresses. Upon further investigation of packet 603's link layer properties, a different MAC address (f0:2c:0f:b2:08:52) is associated with 10.23.0.151, indicating that the user may be attempting to masquerade as the legitimate DHCP server through a false server as we know the original DHCP server was exhausted so it was not able to assign Ip addresses. As a result, f0:2c:0f:b2:08:52 sets up a rogue DHCP server(SPOOFING), which assigns ee:fd:34:6b:72:1c a malicious gateway and IP address 10.23.0.249. The suspicion is confirmed in packet 604 when ee:fd:34:6b:72:1c accepts the offer, and packet 605 acknowledges it and router declines it (606). As a result, any traffic sent by ee:fd:34:6b:72:1c is first sent to the attacker's (f0:2c:0f:b2:08:52) gateway before being sent to the actual router gateway (10.23.0.1).

Recording 2:

In packet 607, the device with MAC address ee:fd:34:6b:72:1c sends an ARP request asking for the MAC address associated with the IP address 10.23.0.151. However, in packets 607 and 608, the device with MAC address f0:2c:0f:b2:08:52 responds with a false MAC address for IP 10.23.0.151, indicating the presence of ARP poisoning. The device with MAC address zyxelcom_41:80:dc then sends another ARP request to verify the MAC address associated with 10.23.0.151, and the attacker responds with their own MAC address in ARP packets 611 and 612, indicating that they have successfully altered the ARP cache table for the router. This allows the attacker to intercept any data sent between the legitimate user and the router, as both the router's and the user's ARP cache tables have been tampered with, as observed in packets 608 and 612. This three-step attack begins with DHCP Exhaustion then a rogue server to act as a candidate for assigning IP addresses, which then allows them to carry out the ARP attack. Packet 615 provides clear evidence that the attacker is not forwarding information to the router, as the legitimate user's attempt to connect to the actual router at 10.23.0.1 is blocked by the attacker, resulting in no response from the router. This confirms that the attacker is intercepting and controlling the network traffic. Packets 617, 618, 619, and 620 show the attacker attempting to update their ARP cache table in order to forward information between the user and the router, and to intercept any data sent between them. Wireshark detects duplicate IP addresses, indicating that multiple MAC addresses are claiming to have the Ip address 10.23.0.151 evidence of Arp cache poisoning.

The attacker first carries out an DHCP exhaustion attack which later stop the router from assigning Ip addresses, this sets up a rogue server attack - assigning the user a malicious gateway, the attacker then tries performing an additional attack known as ARP poisoning to alter the Arp cache table for the users on the network. The attacker did this to increase their chances of intercepting data as if the network has defences for ARP poisoning the rogue server attack would still be carried out and vice versa. The attacker then seems to alter information in the HTTP protocols evidence later in the report. Overall, this is clear evidence of cyber-attack.

The Attack

MAN IN THE MIDDLE ATTACK

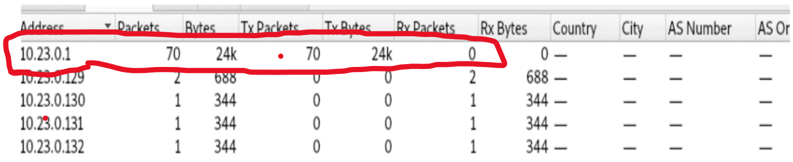
The attack consisted of four parts; one to exhaust the DHCP server, one to setup a rogue server to pretend to be the real router. One to poison the ARP cache table (ARP poison attack) and the other to change credentials of user through intercepting packets sent from user to router and vice versa (HTTP packets).

Reference:

Router:
Mac: b8:d5:26:41:80:d2(**zyxelcom_41:80:dc**)
Ip: 10.23.0.1
Remote server:
Rogue DHCP server (false claiming to have IP 10.23.0.151, set up at MAC: f0:2c:0f:b2:08:52)
Attacker: MICHAEL – exhausted DHCP server
Mac: f0:2c:0f:b2:08:52
Ip: 10.23.0.151
Victim 1:   Gabriel
Mac: ee:fd:34:6b:72:1c
Ip: 10.23.0.249
Victim 2:
Mac: f0:49:41:83:63:8c
Ip: 10.23.0.151 (Mac updated to attacker after ARP spoofing and poisoning of Arp cache table)
City University:
IP: 138.40.78.88
Source: router mac address as it goes through routers default gateway.

Stage 1:

The attacker first floods the network with DHCP requests. This is evidence from packet 6 to 606 and screenshot below, as many DHCP offers were made and acknowledged by the router through the same host name Michael. This is known as a DHCP starvation attack. This allows the attacker to perform stage 2 and 3 of the attack as the router is unable to assign Ip addresses due to reaching Ip limit. This attack is also supported by the fact that no one was replying to ARP messages for having the Ip addresses(packet 350- 352, 329 – 331, 312- 314) the router asks who has an Ip address 10.23.0.152 but there is no response this happens throughout until router is out of Ip address packet 606 DHCP NAK. This is clear evidence of fake MAC users requesting for Ip address.(packet 481, 484, 493 example of starvation there are multiple DHCP offers from router to same host name)



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Or
10.23.0.1	70	24k	70	24k	0	0	—	—	—	—
10.23.0.129	2	688	0	0	2	688	—	—	—	—
10.23.0.130	1	344	0	0	1	344	—	—	—	—
10.23.0.131	1	344	0	0	1	344	—	—	—	—
10.23.0.132	1	344	0	0	1	344	—	—	—	—

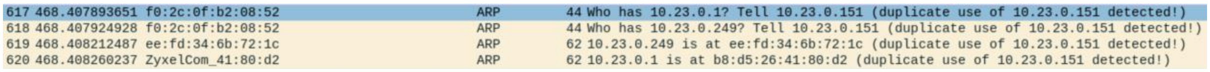
DHCP exhaustion evidence:10.23.0.1 makes the most DHCP calls, clear evidence of Starvation without any response heard back from ARP packets.

Stage 2:

Based on the given information, it appears that an attacker has carried out a man-in-the-middle attack on victim 1 by setting up a rogue DHCP server(DHCP SPOOFING). In this attack, the attacker tricks victim 1 into believing that their machine is the DHCP server by assigning victim 1 a malicious IP address (10.23.0.249) and default gateway (attacker's host machine) instead of the router's default gateway. As a result, any information relayed by victim 1 is first sent to the attacker's machine before being passed on to its intended destination. This allows the attacker to view and modify the information before it reaches its intended destination. This attack is commonly referred to as a rogue server attack and is a type of man-in-the-middle attack. The consequences of such an attack can be severe as the attacker can steal sensitive information or modify it for their own purposes as evidence from recording 1 and stage 1 of attack which made stage 2 possible and easier to execute.

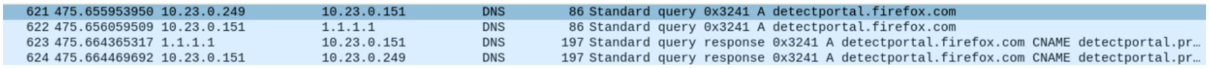
Stage 3:

This rogue server attack is followed by an ARP attack this is to increase the attacker chances of intercepting data as the rogue server attack could fail. The Second stage of the attack consists of poisoning the ARP table by using the IP addresses "10.23.0.151" for multiple Mac addresses. Originally victim 2 was assigned that Ip address. This later is altered, and the attacker resembles themselves as "10.23.0.151" this is clear evidence from packets 608,612,617,618,619,620, 817 and 1230 where duplicate use of the Ip is detected. The attacker successfully alters ARP table to redirect traffic to themselves. This is resembled in recording 2*. The attacker updates the Routers ARP table as well as victim 1 ARP table to f0:2c:0f:b2:08:52 which allows them to forward the traffic to attacker host machine to make it seem normal for victim 1.



```
617 468.407893651 f0:2c:0f:b2:08:52          ARP      44 Who has 10.23.0.1? Tell 10.23.0.151 (duplicate use of 10.23.0.151 detected!)
618 468.407924928 f0:2c:0f:b2:08:52          ARP      44 Who has 10.23.0.249? Tell 10.23.0.151 (duplicate use of 10.23.0.151 detected!)
619 468.408212487 ee:fd:34:6b:72:1c          ARP      62 10.23.0.249 is at ee:fd:34:6b:72:1c (duplicate use of 10.23.0.151 detected!)
620 468.408260237 ZyxelCom_41:80:d2          ARP      62 10.23.0.1 is at b8:d5:26:41:80:d2 (duplicate use of 10.23.0.151 detected!)
```

While Victim 1 uses the network all their data is relayed to the attacker. The attacker is able to intercept data and change it to their liking. Packets 621 622 623 and 624 are evidence of this. The user at Ip address: 10.23.0.249 accesses a DNS server (web page) and the information is not sent straight to the destination (1.1.1.1) instead it is relayed to 10.23.0.151 and then to 1.1.1.1. Data passing is viewed by the attacker and when data is sent back it is sent to the attacker machine first then to the legitimate user. It is very likely that the attacker is forwarding the information to the routers gateway after snooping for information.



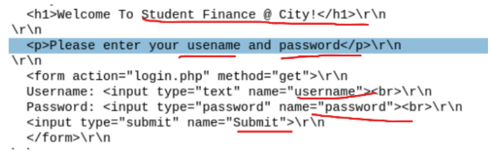
```
  621 475.655953950 10.23.0.249      10.23.0.151     DNS      86 Standard query 0x3241 A detectportal.firefox.com
  622 475.656059509 10.23.0.151      1.1.1.1         DNS      86 Standard query 0x3241 A detectportal.firefox.com
  623 475.664365317 1.1.1.1          10.23.0.151     DNS     197 Standard query response 0x3241 A detectportal.firefox.com CNAME detectportal.pr…
  624 475.664469692 10.23.0.151      10.23.0.249     DNS     197 Standard query response 0x3241 A detectportal.firefox.com CNAME detectportal.pr…
```

HTTP TCP DNS Protocol Suspicious Behaviour (Stage 4)

The attacker only allows victim 1 to send traffic to their machine before its forwarded. This is evidence from packets 621 to 685 all traffic is viewed by attacker first then relayed forward. The three-way handshake with city university was performed by the attacker instead of victim 1(as the victim was trying to access the city website packet691,704-707) resulting in the attacker forming a reliable connection with city university's ensuring all information was directly send to the attacker's machine first as evidenced from TCP/ TLS 2.0 packets from 710 to 735. TLS connection between two hosts with IP addresses 10.23.0.151 and 138.40.78.88. The log shows the exchange of several TCP and TLS packets, including a SYN packet from the client to initiate the connection, SYN-ACK packet from the server to acknowledge the client's request and send its own SYN packet, and ACK packet from the client to acknowledge the server's SYN-ACK packet. Following this, the client sends a TLS Client Hello packet, to which the server responds with a TLS Server Hello packet, and several TLS packets containing certificate, server key exchange, and server hello done messages. Finally, the client sends a TLS Client Key Exchange packet to complete the handshake and establish the secure connection. Three-way handshake ensures the connection is reliable and both clients are ready to send and receive data without it, data could be lost or corrupted.

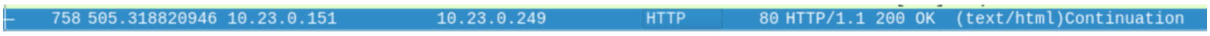
Packet 736 to 769 it looks like the host with IP address 10.23.0.151 is sending data to the host with IP address 10.23.0.249 over port 80, and the data is being acknowledged by the receiving host with malicious port numbers which continuously change throughout. This is clear evidence of data being sent to attacker's machine and attacker decides what to do with it. What happens next is listed below, the victim logins with their credentials and attacker is able to view private credentials.

Using the 'http' filter in Wireshark I was able to narrow down all the HTTP packets. In total it shows 18 http packets. Below is evidence of the intruder trying malicious activity:



```
  <h1>Welcome To Student Finance @ City!</h1>\r\n
\r\n
  <p>Please enter your usename and password</p>\r\n
\r\n
  <form action="login.php" method="get">\r\n
  Username: <input type="text" name="username"><br>\r\n
  Password: <input type="password" name="password"><br>\r\n
  <input type="submit" name="Submit">\r\n
  </form>\r\n
```

In packet 702 we can see the victim trying to access student finance site. A response is given in packet 758 by an HTTP protocol where the website tells victim 1 to enter their credentials. This is obviously relayed to the attacker first so they can do their snooping before sent to either the city university or victim 1.



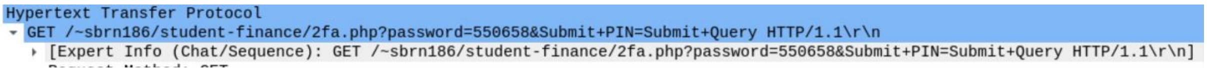
```
   758 505.318820946 10.23.0.151        10.23.0.249        HTTP       80 HTTP/1.1 200 OK  (text/html)Continuation
```

Victim 1 request an favicon.ico file but web page gives a 404 error at packet 811. This could be on purpose by the attacker.

Packet 823 I can observe that credential information such as username and password was entered by victim 1 and passed forward to the attacker. Which relays it the city university server. The attacker is able to view all data as it is not encrypted(HTTP).

Username: root   Password: 04ecc1b53341ebe6

After, conducting a thorough inspection of the packet data, it is evident that the Victim 1 was prompted with a two-factor authentication (2FA) prompt upon logging in frame. The code was sent to the victim's phone number, xxxxx-xxx-689. However, despite the presence of 2FA, the attacker would just relay the information to victim 1 making it seem normal which allows the victim 1 to login to city system.



```
Hypertext Transfer Protocol
▾ GET /~sbrn186/student-finance/2fa.php?password=550658&Submit+PIN=Submit+Query HTTP/1.1\r\n
  ▸ [Expert Info (Chat/Sequence): GET /~sbrn186/student-finance/2fa.php?password=550658&Submit+PIN=Submit+Query HTTP/1.1\r\n]
```

Packet 962 is where victim enters the pin and proceeds to login. The traffic is viewed by the attacker and forwarded. From inspection I can see the 2fa password was 550658. Looking at application layer(http) in detail is it clear when the response packet was received in packet frame 989. All requests sent from victim are relayed to attacker first, the trend seems to follow throughout the packets captured.

In packet 989 it is visible that login was successful, victim was greeted with “welcome to student finance”.

Packet 1000 and 1025 is where the attack is carried out. Victim 1 opens the console for student finance (1000). After closing the console(1025) the details were changed and grants/loans were applied. This was possible due to man in the middle attack. When the victim pressed ok, a request was made to city university to close it, but attacker was able to intercept and type console command before forwarding it. Packet 1025 is clear evidence. Inspecting the application layer further I can identify the commands used to alter personal details below is an image illustration of what was changed in the victim’s city finance account.

Attacker checks the balance changes the personal and financial details (email or password/ name) applies for grant and loan. The victim had no clue about this as everything was behaving normally(1025).

In packet 1119 victim 1 decides to logout in packet/frame 1130 the attacker tells the victim they have successfully logged out without them knowing their details were modified. TCP and TLS v1.2 traffic is captured after which is a normal process to let the city university know that it is done using the website and the secure connection can be disconnected. Packet 1203 is evidence of victim 1 closing their connection. After this there isn’t any data or traffic sent or received from victim 1.

The attacker snoops to find more victims but is unable to do so. From the above findings it has been cleared that the credentials information was changed for victim 1 through series of attack set ups.

Prevention

To ensure the network is secure, it is important to Implement various feature such as encryption, DHCP limit and static Arp tables, the network team should also consider that it shouldn’t present false alarms or block legitimate users from the network. more information about how to prevent this attack are listed below.

To prevent stage 1 DHCP Starvation attacks, several measures can be taken. DHCP snooping is a security feature that can be implemented to monitor DHCP traffic and prevent unauthorized DHCP servers from responding to DHCP requests. Limiting DHCP lease time and implementing. As well as DHCP rate limiting can also help prevent the exhaustion. Additionally, enabling DHCP client verification can ensure that only authenticated clients are allowed to obtain IP addresses this will stop any intruders from joining the network and being assigned an IP address. Finally, segmenting the network into smaller subnets can limit the number of clients that can obtain IP addresses from a DHCP server. By using these precautions, the following packets would have been detected 6 - 606 where the DHCP is exhausted continuously, and no response is given from ARP. This would have reduced the chances of stage 2 and 3 of the attacks.

Preventing Rogue Server Attacks: In a rogue server attack, an attacker sets up a malicious server on the network, posing as a legitimate server and stealing sensitive data. To prevent this, we could implement network segmentation, which involves dividing the network into smaller subnets and isolating servers and devices that require high security. Additionally, implementing port security can prevent rogue servers from being connected to network switches, by ensuring that only authorised MAC addresses are allowed to access the network this helps to prevent stage 2 of the attack as the attacker wouldn’t be able to set up a server to trick the user into being assigned a malicious gateway. Preventing packet 603 604 which would prevent stage 4 of the attack from occurring as the attacker wouldn’t be able to intercept information.

Preventing ARP attacks: ARP attack involves an attacker sending fake ARP messages to associate their MAC address with the IP address of a legitimate device on the network. To prevent this, we could implement ARP spoofing detection tools such as ARP watch, which can detect when an ARP spoofing attack is taking place and alert network administrators. Another method is to use static ARP entries, which manually map MAC addresses to IP addresses and prevent attackers from modifying the ARP cache. This would help resolve phase 3 of the attack as attacker wouldn’t be able to corrupt and poison the cache table in the following packets 616, 617, 618, 619 and stopping the attacker being the MITM, preventing stage 4 of the attack.

Viewing HTTP Data: Stage 4 of an attack involves viewing HTTP data, which is typically unencrypted and can be easily intercepted and read by attackers. To prevent this, we could implement HTTPS, which encrypts HTTP traffic and prevents attackers from viewing sensitive data. We can also implement network security tools such as intrusion detection and prevention systems (IDS/IPS), which can detect and block malicious traffic and alert network administrators in case of a potential attack. Implementing these features would have encrypted the victim’s username and password at packet 823 and preventing the attacker from seeing what the victim accessed at packet 1025 as the attacker wouldn’t of realised what the user was doing through encryption. This would entirely stop the attacker from modifying details.

Overall, it is clear that the attacker successfully altered the victims financial information this was performed is series of step first exhausting the DHCP server then carrying out an rogue server attack followed by an ARP poisoning which later allows the attacker to intercept information and modify it

Report by: Davinder Singh(210022278)