Mærsk Mc-Kinney Møller Institute
Drone centre

Security in Computer systems

Project description

ARP Spoofing: Attack & Defend

Author
Thor Gunnlaugsson Jensen

Teachers
Jan-Matthias Braun
Mathias Neerup
Leon Bonde
Tórur Andreassen
John Hallam

Introduction

This project is about different exploits in the Address Resolution Protocol(ARP). The idea of the this project is to shed light on to this protocol and how the exploits works. This report documents some of the exploits which can be used by an attacker and how a defender defends against those types of attacks and exploits.

1 Background information

The exploits described in this report all starts with the ARP and its inherent weakness'. The ARP is said to be located on the data link layer and used to bridge to the network layer in the OSI[1] model. The 2nd layer in the OSI model which is the layer in charge of node-to-node data transfer and detection and possibly corrections of errors from the 1st layer, the physical layer.

The 3rd layer is responsible of transferring packages between different nodes. Possible by routing packages through other intermidiate nodes.

ARP lies somewhere between the data link layer and the network layer in the OSI model and is a communication protocol used for discovering addresses like MAC addresses which is associated with a IPv4 address typically.

ARP is named so because of the way the address is resolved. The address is resolved by having a client sending a piece of information to the server which uses this information to uniquly indentify this client with that address.

An example with a ARP request and response gives a better understanding[2]:
The ARP request message **"who is X.X.X.X tell Y.Y.Y.Y"**, where X.X.X.X and Y.Y.Y.Y are IP addresses

The target system forms an ARP response **"X.X.X.X is hh:hh:hh:hh:hh:hh"**, where hh:hh:hh:hh:hh:hh is the Ethernet source address of the computer with the IP address of X.X.X.X.

The ARP request is broadcasted to everyone on the LAN however only the client with the IP address in question responds. everyone else silently drops the package.

Since the ARP request also includes the address of the Ethernet source the system who was first asked by an ARP request does not need to send ARP request back.

Any ARP attack is done on a *Local Area Network* (LAN) and works by tampering with the contents of the packages or stopping the traffic to or from some addresses. That is the reason for an ARP attack often is as an opening for other attacks like *Denial of Service*(DoS) or *Man In The Middle* (MITM).

2 Type of attacks

An attack directed at the ARP protocol has several names. The most common name for this type of attack are the ones listed below:

- ARP Poising
- ARP Spoofing

[1] `https://www.webopedia.com/quick_ref/OSI_Layers.asp`
[2] `https://erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html`

- ARP Cache poising

The general idea of ARP attack is to send a spoofed ARP message onto a network and associate the attackers MAC address with the victims IP address. This "wrongly" association made by the host will cause traffic meant for the victim to be send to the attacker. The general idea can be seen in figure 1[3].

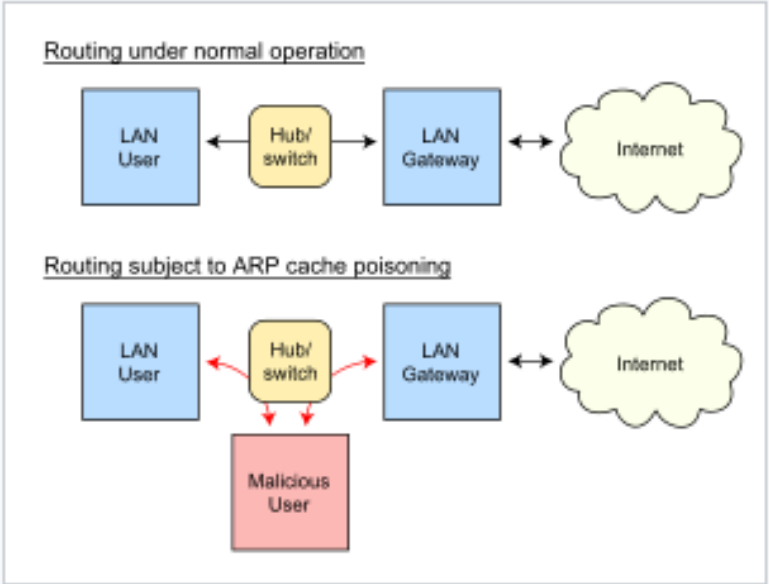


Figure 1: *The top figure show a regular package redirection from the internet to the default gateway and then to client on the local network. The bottom figure shows the traffic being intercepted by an attacker.*

There are several vulnerabilities within the ARP which is exploited. One of them is that the host does not perform any check if it itself send out an ARP-request whenever it receives an ARP-reply.

The host will automatically cache a finite amount of ARP for a finite time. The host does however not check if an ARP entry has expired if the host receives a anorther ARP-reply for the samme IP. A new ARP-reply would make the host update its ARP entries without performing any checks.

2.1 Denial of Service (DoS)

The DoS attack using ARP is a fairly simple attack in the sense that i does not require much extra understanding once the ARP protocol itself is understood. The attack itself consist of sending ARP-reply to the default gateway which then spreads the updated (now tampered) ARP entry to all connected devices. The traffic meant for the victim is now routed to the attacker. If said attacker does not chose to forward the traffic to the victim the attacker has performed a DoS attack since the victim receives no traffic even though it was requested.

[3] `https://en.wikipedia.org/wiki/ARP_spoofing`

192.168.38.113	00:24:D7:04:5A:30
192.168.38.108	00:90:A9:B2:62:42
192.168.38.1	34:21:09:24:AE:B8
192.168.38.110	B8:27:EB:A4:3B:06
192.168.38.103	C4:61:8B:82:5E:D0
192.168.38.106	D0:4F:7E:0B:CC:0D

(a) *This figure lists all the devices found on the local area network and their associated hardware addresses. The one marked orange is the intended victim.*

```
wlo1      Link encap:Ethernet  HWaddr 00:24:d7:04:5a:30
          inet addr:192.168.38.113  Bcast:192.168.38.255
```

(b) *This is the interface device, IP address and hardware address taken directly from the victim device.*

Figure 2: *More than the victim device is picked up by Ettercap on the LAN. We see that the information found via Ettercap is correct when comparing it to the information taking directly from the victim.*

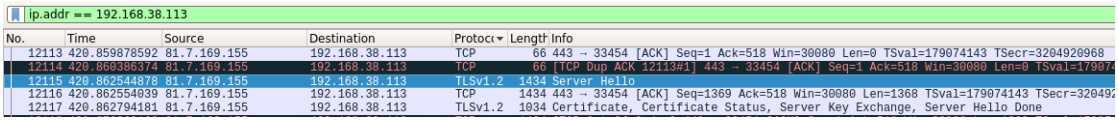


Figure 3: *A filter is applied in Wireshark to only show the packages meant for IP address* ***192.168.38.113*** *which is the victim. We see from this cutout that the victim has entered a web page and recieved a* ***"server hello".***

Another way of using the power of redirecting traffic is to flood a victim with traffic rendering the victim unable to do any other task. This can be done if the attacker takes all the IP addresses and associate one machines MAC address to all of them. The combined traffic would then "flood" the victim, again, rendering the victim unable to do any other task.

In order to get a deeper understanding this type of attack the DoS was investigated in real life on the authors own machines. The first thing attack done was not a DoS but only a eavesdropper attack done to see the apparent vulnerability of ARP firsthand. The tool Ettercap[4] was used to poison the ARP entries and associate the attacking device to the victims IP hardware address. The target IP address and hardware address found by Ettercap can be seen in figure 2a, and the actual IP address and hardware address is seen in figure 2b. It is a match. Wireshark[5] was then started to capture the traffic when the victim entered a web page as is seen in figure 3.

Next an attempt to do a MAC flooding was tried. The idea with this attack was to flood a table with random MAC addresses forcing the switch to broadcast the traffic onto the network instead of directing it. This was done using the tool macof and a directioned attack on the victim through a terminal as so:

```
sudo macof -i eno1 -n 100000000 -d 192.168.38.113
```

[4] `http://www.ettercap-project.org/ettercap/`
[5] `https://www.wireshark.org/`

where *-i eno1* is the network interface, *-n 100000000* is the number of messages and last the victims IP address. Figure 4 shows the unfiltered output of Wireshark where the LAN is being flooded by packages to the victim. The author would also point out that the victim were unable to browse any web pages while the attack was underway and as the attack kept progressing other devices on the authors LAN start to have difficulty in browsing online.

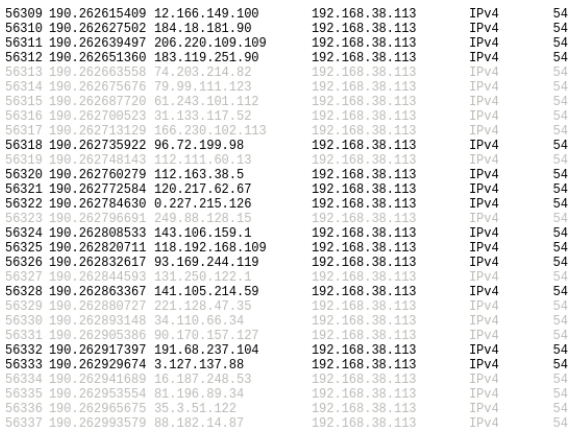


Figure 4: *This screen-shot from Wireshark show the network being flooded with messages to the victim.*

2.2 Man-In-The-Middle (MITM)

ARP can also be exploited performed to do MITM attacks. The attacker can now alter the information to and from the desired end destination. This attack gives the impression that the two affected parties, victim and the end destination are talking between themselves even though an attacker is sitting in the middle. ARP spoofing is more of a passive attack where the attacker listens to the traffic flowing through but ARP spoof combined with another attack is very potent and opens the possibility of active attacks instead passive.

Another type of MITM attack is DNS spoof where the attacker redirects the victim onto a page of the attackers choosing while tricking the victim into still believing he is on the correct page.

This attack was also tested using Ettercap. This test was designed to show how an attacker could fool a victim into joining a dangerous network by spoofing the DNS. First Ettercap is launched and the attacker starts sniffing the network by choosing which network interface to use as seen in figure 5.

Then the attacker needs to scan for hosts to do the MITM attack. The available hosts are seen in figure 2a.

The host chosen to attack is the same victim as before **192.168.38.113** and the switch **192.168.38.1**. Next the attacker has load the DNS spoof plugin into Ettercap but first the attacker has to create or choose where to route the victim. If the attacker want to keep a low profile only the particular names are routed to the dangerous site. In this example only the Nordea banks website is spoofed. This is done by altering the DNS spoofs plugin file located either under */usr/share/ettercap/etter.dns* or */etc/ettercap/etter.dns*. The alterations can be

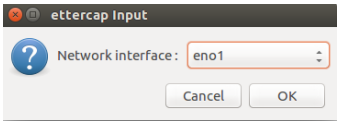
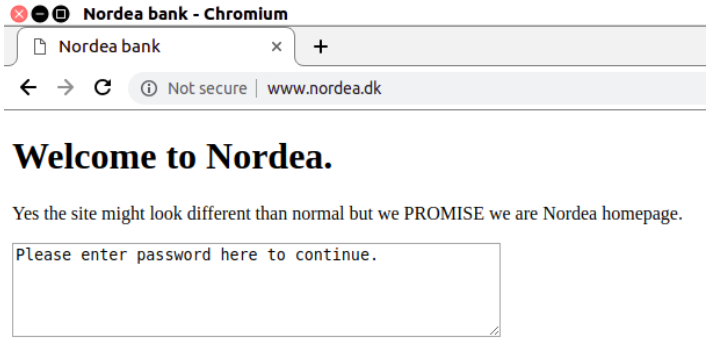


Figure 5: *This window lets the user chose with interface to use Ettercap with.*

```
############################## has to cre
# For SCS project - plz delete afterwards
*.nordea.com A 192.168.38.105
*.nordea.dk A 192.168.38.105

www.nordea.com PTR 192.168.38.105
www.nordea.dk PTR 192.168.38.105
```

Figure 6: *This screen dump shows which alterations there were to Ettercaps DNS spoof plugin. Observe that the American and Danish domain of Nordea are mapped to the attackers computer which is running evil copy of Nordeas homepage.*



(a) *Caption*

```
tgj@tgj-HP-ProBook-6560b:~$ ping www.nordea.dk
PING www.nordea.dk (192.168.38.105) 56(84) bytes of data.
64 bytes from www.nordea.dk (192.168.38.105): icmp_seq=1 ttl=64 time=10.8 ms
64 bytes from www.nordea.dk (192.168.38.105): icmp_seq=2 ttl=64 time=3.04 ms
64 bytes from www.nordea.dk (192.168.38.105): icmp_seq=3 ttl=64 time=3.59 ms
64 bytes from www.nordea.dk (192.168.38.105): icmp_seq=4 ttl=64 time=12.4 ms
```

(b) *Caption*

Figure 7: *The figures above is proof that the DNS spoof attack worked and the victim device was tricked into believing that Nordeas homepage was actually located at IP address* ***192.168.38.105*** *which is the attackers IP.*

seen in figure 6

The attack outcome of the attack can be seen in figure 7. One may notice that figure 7a has a strikingly resemblance to Nordeas actual homepage. Also if the victim tried to ping Nordeas homepage, it would receive a response as seen in figure 7b. Creating an elaborate working duplicate of the Nordeas homepage was out of scope for this report.

3 Defence/Countermeasure

Use as little trust as possible. The implied trust between machines in a network makes it possible for a attacker to spoof a trusted machine. One of the major drawbacks of ARP is that there is no authorization that control whether or not the ARP reply is correct or even requested by the switch.

There is no catch all solution but it is possible to be proactive and/or reactive if one suspects ARP poisoning.

One solotion is to hardcode the ARP table into each machine on the network. This require that the network is more or less a static network otherwise it would be a lot of trouble to update this with every new device joining the network. The smart thing about this is that it does not rely on ARP request reply. The downside is that it is a hazzle to add and remove devices.

The reactive approach requires the administrator to use a third party software to monitor the network monitor of hosts. This involves looking at package patterns to see if their are any irregularities. This will require that the administrator has some insight into when and where the data flows in his networks.

4 Conclusion

One of the main problems with ARP is the lack of authentication. Anyone on the network can send a ARP message which is serious security risk because an attacker can associate his MAC address with the victims IP address.

ARP poisoning is effective and subtle which makes it hard to protect against. ARP poisoning can be very dangerous combined with another attack/exploit.

5 Reflections

When the author did the MITM attack the victim device was trying to access Nordeas homepage through the Firefox browser. Firefox did not allow the victim device to enter the address spoofed from the attacker. However when the victim switched to Chromimium browser the victim was allowed to enter the attackers web page.
This caused some timedelay and some serious headscrathing for the author.

When researching how to defend against ARP spoofing it came apparent that it was easier said than done. If one chose a proactive stance one would end up with a rigid network which would take a lot more maintenance from the administrator.
A more reactive stance would involve installing a third party software which also requires maintenance and supervision. Also this solution assumes the attack is in progress.