Group 38's Report

Daan Boelhouwers (1457152)
d.boelhouwers@student.tue.nl

Richard Farla (1420380)
r.farla@student.tue.nl

Ivar de Win (1406663)
i.j.f.d.win@student.tue.nl

June 18, 2022

1 Introduction

This report describes group 38's implementation of the default project. For the default project, we are required to create a fully-fledged poisoning/spoofing tool for ARP and DNS with SSL stripping capabilities. Here fully-fledged means that the tool; is fully automated, adapts to scenarios with different IPs and numbers of hosts, has a silent and all out mode, and requires a minimal manual to use.
For a short demo you can watch this video.
The project itself can be found here.

2 Attack Description

Our tool supports ARP and DNS spoofing. These techniques can be used for man in the middle (MITM) attacks. During such an attack, the attacker intercepts messages between a user and an application. This allows the attacker to either eavesdrop on the messages, or impersonate one of the parties, making it appear as if a normal exchange of information is underway [1].

2.1 ARP Spoofing

ARP spoofing is an attack where the attacker sends forged ARP response messages onto a network. The forged messages advertise that a victim's IP address belongs to the attacker's MAC address. Other devices will update their ARP cache to match the content from the forged messages. These devices now send all packets meant for the victim to the attacker instead, thus a MITM position has been acquired.

The ARP protocol does not verify whether a response to an ARP request really comes from an authorized party. The protocol also allows devices to accept ARP responses within having sent a request first. These vulnerabilities allow ARP spoofing attacks to be effective, as the attacker can easily pretend to be someone else and does not have to wait for the perfect timing. The newer NDP protocol addresses this issue, but is not widely in use yet [2].

2.2 DNS Spoofing

There are two kinds of DNS spoofing attacks. For the first, an attacker injects fake DNS entries into a DNS server. The fake entries map the domain name of an existing website to an IP address that the attacker provided. Whenever a user then attempts to access a URL that belongs to the spoofed domain name, they are redirected to the attacker's website instead.

For the type of DNS spoofing we implemented, the attacker intercepts the communication between a victim and a DNS server. The attacker then replies to the victim with a malicious IP address instead of the IP address that belongs to the URL that the victim tried to access. It is important that the attacker is already in a MITM position for this attack to be successful. As such, ARP spoofing should be performed before DNS spoofing [3].

3 Technical Setup

To perform the attacks, it is important that the attacker machine has a Linux operating system, preferably Ubuntu. The machine must be able to install Python 3, pip, and Scapy. Having them pre-installed will work as well. The machine must be connected to the network on which the attacks are planned to be executed, in order to reach other devices.

In our case, this setup was done using two of the virtual machines provided by the university, and an extra virtual machine to perform the attack. The extra virtual machine is called *Ubuntu 22.04*, as it uses a (64-bit) Ubuntu version 22.04 operating system. This machine conforms to the requirements stated above, as it is used to perform the attack.

The *M3 Linux Attacker* machine uses a (64-bit) Linux Mint 18.3 "Sylvia" operating system. The university initially intended this machine to be the attacker, but we ran into issues preparing the machine for the attack. As such, we introduced the *Ubuntu 22.04* machine to perform the attack instead. Since we still wanted to use the *M3 Linux Attacker* machine, we turned it into our victim.

The virtual machine *M2 Server* has a (64-bit) Oracle operating system and functions as the server to which the victim tries to connect. Even though a machine called *M1 XP* is shown in Figure 1, it is not used for our attack.

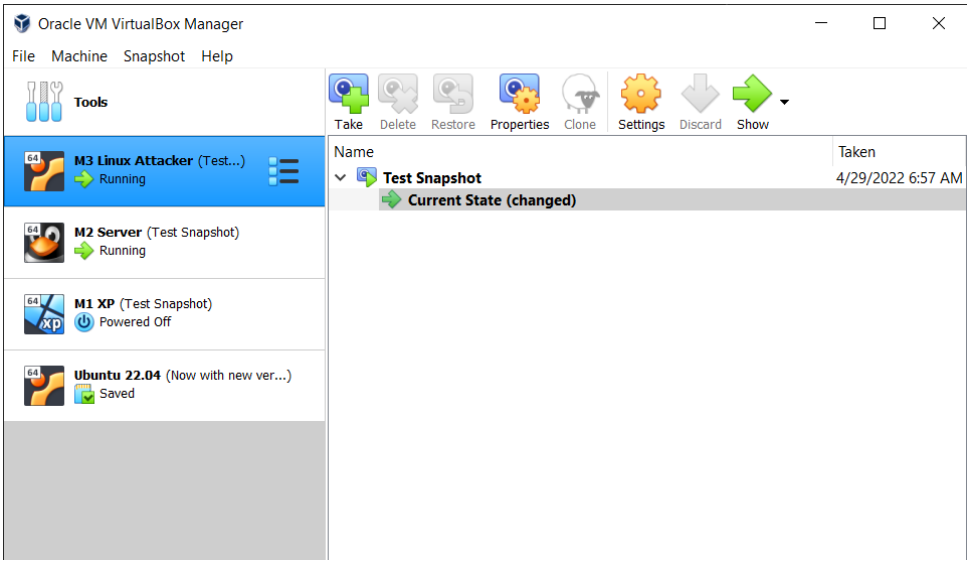

Figure 1: The virtual machines

4 Attack Analysis

This section will show a usecase for both ARP Spoofing and DNS spoofing.

4.1 ARP Spoofing

Person A is an owner of a home. Person A has decided to rent out this home to person B. However, after a period of time has passed, person A notices that person B has a relatively high internet usage. Person A would like to know if there is any malicious activity going on in the house that he has rented out.
Hence, person A decides to set up a device in the home. This device will connect to the main network, and will perform ARP spoofing on this network. Now, the device will be a Man in the Middle for all devices on this network. The device will forward all connections such that person B will still be able to use the internet as normal. As a result, the device is able to track

the internet usage of person B. Hence, person A is able to detect any malicious internet usage of person B.
Person A is able to find out with this method that person B does indeed connect to websites that trade in illegal goods. Hence, person A is able to report person B to the police, and will be able to avoid any legal liability.
The given scenario shows a potential usecase of ARP spoofing. ARP spoofing is able to create a Man in the Middle scenario, which allows for the tracking of internet usage. Without this method of ARP spoofing, tracking internet usage might potentially be much more difficult.

4.2 DNS Spoofing

Person A is in a situation where person B, an acquaintance of person A, has managed to steal an internet account of person A. Person A is unable to recover this account by conventional means. Person A would like to know the new credentials of this account in order to regain access to this account. Hence, person A decides to try DNS spoofing.
Person A first becomes a Man in the Middle, using ARP spoofing, in order to be able to DNS spoof B. Once A is a Man in the Middle, A will forward all traffic to their intended locations except for traffic that are a DNS request for the IP-address corresponding the the website associated to the stolen account. Person A will respond to this DNS request with a spoofed DNS response containing an incorrect malicious IP-address. Now, when person B tries to connect to the website, person B will actually connect to a incorrect IP-address. Person A is able to host a look-alike website on this IP-address, where person B is likely to input the credentials of the account. Hence, person A is able to regain access to the account with DNS spoofing.
The given scenario shows a potential usecase of DNS spoofing. DNS spoofing is able to poison the DNS cache of a target, which will make the target connect to an incorrect IP-address. This IP-address can host multiple different websites, according to the needs of the person administering the DNS-spoofing.

5 Attack Engineering

The program opens a command line interface to interact with the user to get the information needed to run the program as required. This information includes variables such as the victim ip, the victim MAC, the DNS domain to spoof, the IP and MAC discovery method, etc. The program will use this information to both apply ARP spoofing to a given victim on a given interface to achieve a Man in the Middle status, and apply DNS spoofing as a Man in the Middle attack.
The program starts in the discovery phase. After the user inputs which interface to discover on, the program will either discover IP and MAC address pairs actively or passively on the given interfaces. During passive discovery, the program will sniff the network and report any IP and MAC address pair it finds on the network. During active discovery, however, the program will send an ARP Request for every possible IP in the subnet on every chosen interface, and will report the responses. This will acquire all IP and MAC address pairs on the network, but will send potentially tens of thousands of packets on the network.
The user is able to end the discovery phase at any time. The program will subsequently report all IP and MAC addresses that it has found during the discovery phase. The user will be able to use this information for the next phase of the program.

5.1 ARP Spoofing

In the ARP spoofing phase of the program, the user is able to choose MAC and IP addresses of multiple hosts for which the user wants to be a Man in the Middle. The user has to select at least 2 hosts, since this is the minimum amount of hosts required to set up a Man in the Middle

attack.
All traffic between the selected hosts will be redirected to the program. The program achieves this by sending Spoofed ARP packets to all selected hosts every 3 seconds. A spoofed ARP packet sent to a host will contain the IP addresses of another selected hosts together with the MAC address of the machine on which the program is active. Hence, when a selected hosts tries to communicate with another selected host, the traffic will be directed to the program instead. The program has now achieved a Man in the Middle status.
The program is able to forward all traffic towards its intended destination. The program has the correct information about IP addresses and MAC addresses from the discovery phase, and is thus able to sent all traffic to the correct MAC address corresponding to the IP address which the traffic was intended for. Hence, the selected hosts will still be able to communicate with each other. However, the program is able to analyze the communication between the selected hosts.

5.2 DNS spoofing

After the user has given the program the information related to ARP spoofing, the program will prompt the user to select whether the user wants the program to apply DNS spoofing as well. After the user chooses to apply DNS spoofing, the user can input one or multiple domain names to spoof. The user will subsequently choose an IP to redirect the hosts to when a host executes a DNS request to one of the selected domain names.
After the user has input the required information, the program will start DNS spoofing together with the ARP spoofing. The ARP spoofing allows for the program to be able to inspect the traffic of the hosts for DNS requests. The program is then able to block these DNS requests from reaching their intended destination, and is able to send spoofed DNS responses according to the information given by the user. DNS requests that do not relate to the selected Domain Names will not be blocked or spoofed. Hence, the local DNS cache of a host will be poisoned by the program when this host makes a DNS request to a selected domain name.

6 Future Work

The project required a tool for ARP and DNS poisoning with SSL stripping capabilities. Since the SSL stripping capabilities are not present in our tool, this has a high priority for being added in the future.

Our tool can currently only be successfully used on Linux systems. We would like to expand this to other operating systems as well. This way people will not have to struggle as much with getting a virtual machine to work, which can actually run the tool.

To increase usability even more, we would like to add a graphical user interface to the tool. This is generally a lot nicer to work with than having to set all options via command-line inputs. The use of a GUI could even change the linearity of the application, allowing the user to make modifications to their current attack while it is already running.

References

[1] Imperva. *Man in the middle (MITM) attack*. URL: `https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/`. (accessed: 16.06.2022).

[2] Imperva. *ARP Spoofing*. URL: `https://www.imperva.com/learn/application-security/arp-spoofing/`. (accessed: 16.06.2022).

[3] Imperva. *DNS Spoofing*. URL: `https://www.imperva.com/learn/application-security/dns-spoofing/`. (accessed: 16.06.2022).