Introduction

This course details how to perform a Person-in-the-Middle attack against a client performing an HTTP connection. Knowing how to perform this type of attacks is critical when performing mobile applications testing.

The exercise

In this exercise, you have control over the client's DNS resolution. Using this you will be able to lie and tell the client to go to your server instead of the legitimate one. If you manage to get access to the request's content, you finished this exercise.

This exercise is divided in two steps:

- Setting up a DNS server.
- Setting up a TCP server.

This is an introduction to more complex exercises that will go into details on how to intercept TLS connection.

To follow this exercise, you will need a public IP address with root access. You can either:

- Open your home router and use a Linux/Unix system
- Start a server (for example with DigitalOcean - referral link).

Setting up a DNS server

To setup a quick and easy DNS server, you can use `dnsmasq`. It can easily be installed using `apt`, `brew` or `yum` depending on your system.

Once you have it installed, you can create a very simple configuration file (`dnsmasq.conf`):

addn-hosts=dnsmasq.hosts

This configuration will tell `dnsmasq` where to look for your resolution file (`dnsmasq.hosts`).

The `dnsmasq.hosts` file has a syntax very similar to `/etc/hosts`:

```
1 127.0.0.123   test1.pentesterlab.com
```

Once both files are created, you can start `dnsmasq` in the foreground using:

```
1 $ sudo dnsmasq -C dnsmasq.conf --no-daemon
```

If `dnsmasq` refuses to start it's probably because it may already be running. Some Linux flavors automatically start the daemon when you install it. Make sure you stop the service before running this command.

You can then test your configuration using `dig` or `host`:

- with `dig`:

```
% dig @localhost test1.pentesterlab.com

; <<>> DiG 9.8.3-P1 <<>> @localhost test1.pentesterlab.com
; (3 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15562
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;test1.pentesterlab.com.    IN  A

;; ANSWER SECTION:
**test1.pentesterlab.com. 0 IN  A 127.0.0.123**

;; Query time: 6 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Jan  3 11:11:12 2016
;; MSG SIZE  rcvd: 56
```

- with `host`:

```
% host test1.pentesterlab.com 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:
```

test1.pentesterlab.com has address 127.0.0.123

To perform this attack you will obviously need to know what host the server is trying to access. To do so, you can use `tcpdump`:

```
% sudo tcpdump -i [INT] udp port 53
```

Where:

- `[INT]` is the interface you want to listen on. For example, `eth0`.
- `udp port 53` is used to only sniff UDP packets on port 53 (to keep DNS queries).

After getting the application to use your server as DNS server, you should see a DNS query in `tcpdump`:

```
% sudo tcpdump -i eth0 udp port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol
    decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144
    bytes
05:30:44.362946 IP ptl-intercept-cd787546.libcurl.so.50148 >
    104.131.54.221.domain: 43145+ AAAA? mitm1.pentesterlab.com. (40)
05:30:44.363151 IP ptl-intercept-cd787546.libcurl.so.51919 >
    104.131.54.221.domain: 39139+ A? mitm1.pentesterlab.com. (40)
05:30:44.363716 IP ptl-intercept-cd787546.libcurl.so.43344 >
    104.131.54.221.domain: 30919+ A? mitm1.pentesterlab.com. (40)
```

Based on this information, you can update your `dnsmasq.hosts` file.

A simple TCP server

Know that we got the client to connect to us, we just need to setup a TCP server. This can easily be done using:

- `netcat` with `sudo nc -l -p 80`
- or `socat`:

```
$ socat TCP4-LISTEN:80,fork,reuseaddr -
```

Conclusion

This exercise showed you how you can intercept a TCP connection from a client when you have control over its DNS resolution. Using that, it's trivial to gain access to the information transmitted over insecure communications. I hope you enjoyed learning with PentesterLab.