Lab: Man in the Middle Attacks

6COSC002W

Ayman El Hajjar

Week 7

Requirements and Notes

- **NOTE**- In some activities you need **only** your Kali Linux machine to be connected to the internet.
- In this lab, you need the following VM machines
  1. Kali Linux
  2. OWASP Vulnerable machine
  3. Windows 7 or windows 10 virtual Machine - Check the lab environment setup for more details

Man in the Middle Attacks

A Man in the Middle (MITM) attack is the type of attack in which the attacker sets himself in the middle of the communication line between two parties, usually a client and a server. This is done by breaking the original channel and then intercepting messages from one party and relaying them (sometimes with alterations) to the other.

Setting up a spoofing attack with Ettercap - ARP Spoofing

Address Resolution Protocol (ARP) spoofing is maybe the most common MITM attack out there. It is based on the fact that the Address Resolution Protocol—the one that translates IP addresses to MAC addresses—does not verify the authenticity of the responses that a system receives. This means that, when Alice's computer asks all devices in the network, "what is the MAC address of the machine with IP xxx.xxx.xxx.xxx", it will believe the answer it gets from any device, be it the desired server or not so ARP spoofing or ARP poisoning works by sending lots of ARP responses to both ends of the communications chain, telling each one that the attacker's MAC address corresponds to the IP address of their counterpart.

Note

- In this experiment, it is essential that you identify the IP addresses of both VMs.

- With the virtual machines running, our Kali Linux (192.168.56.101) host will be the attacking machine. Open a root terminal and run the following command:

Start Ettercap

- sudo ettercap –G
- in Ettercap menu, select sniff – unified sniffing
  - You will need to select your virtual interface to sniff on. In my case the virtual interface is vboxnet0
  - This is the virtual network interface you created when you setup your virtual lab environment to a private isolated network.
- Now that we are sniffing the network, the next step is to identify which **hosts** are communicating. To do that, go to Hosts on the main menu, then **Scan for hosts**.
- From the hosts we found, we will select our targets. To do this from the Hosts menu, select Hosts list.
  - From the list, select 192.168.56.102 and click on Add to Target 1.
  - Then, select 192.168.56.103 and click on Add to Target 2.
- Now we will check the targets: on the Targets menu, select Current targets:
  - We are now ready to start the spoofing attack and position ourselves in between the server and the client. From the Mitm menu, select ARP poisoning..
- In the pop up window, check the box Sniff remote connections and click on OK
- And that's it, we can now see all traffic between the client and the server.
- On the victim machine (Windows) - suppose you are visiting a genuine website. Open a browser and visit the vulnerable machine. Dont forget the attacker is using **ARP poisoning attack**.
- log in to the Damn Vulnerable Web Application.
- The attacker should be able to see all traffic. In fact ettercap is able to recognise username and password on a log in page.
- If you return to the attacker machine, you should be able to see the username and password on Ettercap screen.

- Explore the various options using the **man** command to see the additional options. Those options can help if you are looking for using Ettercap in a specific way

Wireshark and MiTM traffic

Ettercap can detect when relevant information such as passwords is transmitted through it. However, it is often not enough to intercept a set of credentials when performing a penetration test, we might be looking for other information like credit card numbers, social security numbers, names, pictures, or documents. It is therefore useful to have a tool that can listen to all the traffic in the network so that we can save and analyze it later; this tool is a sniffer and the best one for our purposes is Wireshark and it is included in Kali Linux..

- Now the setup is done to see the traffic between client and server, how do you actually carry on to see the traffic?

Using Wireshark

- Run Wireshark from the middle of the Windows client and vulnerable_vm from Kali's Applications menu — Sniffing & Spoofing or from the terminal run: **wireshark**
- Make sure you use **eth0** as your network interface. **Click start**
- You should be able to see traffic now.
  - If not, make sure you browse in the client machine (that is your host machine spoofed) to the OWASP vulnerable machine - that is the web server.
  - you should now be able to see genuine traffic between client and web server

Modifying data between the server and the client

So far we have only managed to spoof arp messages and thus sniff all genuine traffic between client and web server but What if we want to intercept messages and tamper with them. That is modifying data such as requests and responses exchanged between client and server.

- We need Ettercap filters to detect whether or not a packet contains the information we are interested in and to trigger the change operations.
- Using Nmap we can find if a host is up or not.
- Nmap has a GUI version called Zenmap, which can be invoked by issuing the command zenmap at the terminal window or by going to Applications — Kali Linux — Information Gathering — Network Scanners — zenmap.

Note

- This experiment will not work unless you have completed the previous experiments and you have spoofed the ARP of the victim.
- You are also sitting in the middle sniffing traffic between client and web server

commands for determining network range

- Our first step is to create a filter file. Save the following code in a text file (we will call it regex-replace-filter.filter) as is shown below.
  - **Note**: The # symbols are comments., The syntax is very similar to C apart from that and a few other little exceptions.

```
# If the packet goes to vulnerable_vm on TCP port 80 (HTTP)
 if (ip.dst == '192.168.56.102'&& tcp.dst == 80) {
# if the packet's data contains a login page
 if (search(DATA.data, "POST")){
msg("POST request");
 if (search(DATA.data, "login.php") ){
msg("Call to login page");
# Will change content's length to prevent server from failing
 pcre_regex(DATA.data, "Content-Length\:\ [0-9]*","Content-Length: 41");
msg("Content Length modified");
# will replace any username by "admin" using a regular expression
 if (pcre_regex(DATA.data, "username=[a-zA-Z]*&","username=admin&"))     {
msg("DATA modified\n");
}
msg("Filter Ran.\n");
}
}
}
```

- Next, we need to compile the filter for Ettercap to use it. From a terminal, run the following command:

Finding open ports

1. etterfilter -o regex-replace-filter.ef regex-replace-filter.filter
2. Now, from Ettercap's menu, select Filters — Load a filter, followed by regex-replace-filter.ef and click Open:
   - We will see a new entry in Ettercap's log window indicating that the new filter has been loaded.
3. In the victim client, browse to http://192.168.56.102/dvwa/ and log in as any user with the password admin , for example: inexistentuser : admin .
4. The user is now logged in as an administrator and the attacker has a password that works for two users.
5. If we check Ettercap's log, we can see all the messages we wrote in code displayed there.

- There is more Ettercap filters can be used for other things besides altering requests and responses, they can be used.
- For example to log all HTTP traffic and execute a program when a packet is captured:

```
if (ip.proto == TCP) {
if (tcp.src == 80 || tcp.dst == 80) {
log(DATA.data, "./http-logfile.log");
exec("./program");
}
}
```

- For more information on Ettercap filters, check out the etterfilter man page.
- Etterfilter is a very powerful tool.

How it works

An ARP spoofing attack is only the start of more complex attacks. In this recipe, we used the packet filtering capability of Ettercap to identify a packet with specific content and modified it to force the user to log in to the application as an administrator. This can also be done from server to client and can be used to trick the user by showing them fake information.
Our first step was to create the filtering script, which first checks if the packet being analysed contains the information that identifies the one we want to alter, as illustrated:

```
if (ip.dst == '192.168.56.102'&& tcp.dst == 80) {
```

If the destination IP is the one of the vulnerable_vm and the destination TCP port is 80 which is the default HTTP port, it is a request to the server we want to intercept.

```
if (search(DATA.data, "POST")){
msg("POST␣request");
if (search(DATA.data, "login.php") ){
```

If the request is by the POST method and goes to the login.php page, it is a login attempt as that is the way our target application receives the login attempts.

```
pcre_regex(DATA.data, "Content-Length\:\␣[0-9]*","Content-Length:␣41");
```

We used a regular expression to locate the Content-Length parameter in the request and replaced its value with 41, which is the length of the packet when we send a login with admin/admin credentials.

```
if (pcre_regex(DATA.data, "username=[a-zA-Z]*&","username=admin&")){
msg("DATA␣modified\n");
}
```

Again, using regular expressions, we look for the username's value in the request and replace it with admin. The messages (msg) are only for tracing and debugging purposes and could be omitted from the script. After writing the script, we compiled it with the etterfilter tool for Ettercap in order to process it. After that, we loaded it into Ettercap and then just waited for the client to connect. There's more... Ettercap filters can be used for other things besides altering requests and responses, they can be used, for example, to log all HTTP traffic and execute a program when a packet is captured:

```
if (ip.proto == TCP) {
if (tcp.src == 80 || tcp.dst == 80) {
log(DATA.data, "./http-logfile.log");
exec("./program");
}
}
```

They also display a message if a password has been intercepted:

```
if (search(DATA.data, "password=")) {
msg("Possible␣password␣found");
```