Exp 06

Matteo Di Fabio: 264339

Luca Genovese: 264364

Introduction:

The goal of this experiment is to manipulate and modify the content of some packets that are exchanged between two computer: in particular, the file that is transferred is a plain text file.
We found 2 different possible approach to conduct this type of attack.
Both methods expect that the attacker will locate itself in the middle of the communication between the 2 points; so the first step is to accomplish successful the man in the middle attack.

The first method consists of creating a desynchronized state on both ends of the TCP connection so that the two points cannot exchange data any longer. A third party host is then used to create acceptable packets for both ends which mimics the real packets.
The term desynchronized state will refer to the connection when both sides are in the ESTABLISHED state and no data is being sent. This state is stable as long as no data is sent.
Here is summarizes the desynchronization process:

- The attacker listens for a SYN/ACK packet from the server to the client stage in the con nection set up
- On detection of that packet the attacker sends the server a RST packet and then a SYN packet with exactly the same parameters TCP port but a different sequence number
- The server will close the RST connection when it receives the RST packet and then reopens a new one on the same port but with a different sequence number on receipt of the SYN packet It sends back a SYN/ACK packet to the client
- On detection of that packet the attacker sends the server a ACK packet The server switches to the ESTABLISHED state
- The client has already switched to the ESTABLISHED state when it receives the rst SYNACK packet from the server

Both end point are in the desynchronized ESTABLISHED state now.
The sequence number from client to server (the one expected from the server) is managed by the server. Instead the sequence number from the server to the client is managed by the attacker. The success of the attack relies on the correct value being chosen for the sequence number for the client (by the attacker). Wrong value may make the client's packet acceptable and can produce unwanted effects.
Note that ACK is the expected SEQ number that the "A" expect to receive as next packet, instead, the sequence number is the number in byte of all the data sent to "B", so each sequence number is the last value plus the dimension of the last sent data. The sequence number is the number of bytes that it is already sent while the ACK is the sequence number that it is expected to be received from the interlocutor namely the other end point.

The second method instead plans to create a man-in-the-middle attack, where all the packets forwarded from the attacker will be filtered (through a proper filter) and consequently modified/manipulated automatically. We decided to use this second approach and we will explain it in the details in the following paragraph.

Injection attack in detail:

To carry out this attack it is used a private network to prepare a proper environment in which 3 virtual machine can communicate each other in host only mode, without any communication with outside internet. Then using 2 scripts (one on the Client machine and one on the Server machine), a file is sent through TCP protocol, from computer_1 (Client) to computer_2 (Server). First of all we try to send the file with a standard communication, without any attacker or man in the middle, and we analyse the traffic and all the packet exchanged on wireshark from which we could extrapolate the following information:

Computer_1 → IP 192.168.168.106, MAC 08:00:27:f0:d6:29
Computer_2 → IP 192.168.168.109, MAC 08:00:27:42:7a:21
Attacker → IP 192.168.168.104, MAC 08:00:27:e7:5a:5b

Then we check the ARP table:

Computer_1

```
luca@luca-VirtualBox:~/Scrivania/Sicurezza_client$ arp
Indirizzo TipoHW IndirizzoHW Flag Maschera Interfaccia
192.168.56.109           ether   08:00:27:42:7a:21   C                     enp0s3
```

Computer_2

```
luca@luca-VirtualBox:~/Scrivania/Sicurezza_server$ arp
Indirizzo TipoHW IndirizzoHW Flag Maschera Interfaccia
192.168.56.100           ether   08:00:27:53:64:7d   C                     enp0s3
192.168.56.106           ether   08:00:27:f0:d6:29   C                     enp0s3
```

Then we write a python script that exploiting the tool ettercap is able to carry out the attack in the following way:

- An ettercap filter is created: in this way through some rule that specified the connection to attack and so on, all the space are substituted by smile (☺).
- A mitm is established and so the attacker machine stands between the 2 end points, in this way, through a socket (wireshark) it is possible to read all the packet that the victims exchanges each other. The mitm is established between 2 victims that the attacker choose writing their proper IP as argument when the script is launched.
- At this point the mitm, using the filter previously created can forward all the packet manipulating the byte and changing the spaces with smiles.

At this point we have observed that in wireshark all packets designated to the IP address of the two computers are indeed sent to the mac address of the attacker and, in turn, it will forward the packets to the correct destination mac address.
This happens because through the script we have done an ARP poisoning as we can see from the following screenshot:

Computer_1:

```
luca@luca-VirtualBox:~/Scrivania/Sicurezza_client$ arp
Indirizzo TipoHW IndirizzoHW Flag Maschera Interfaccia
192.168.56.100           ether   08:00:27:53:64:7d   C                     enp0s3
192.168.56.104           ether   08:00:27:e7:5a:5b   C                     enp0s3
192.168.56.109           ether   08:00:27:e7:5a:5b   C                     enp0s3
```

Computer_2:

```
luca@luca-VirtualBox:~/Scrivania/Sicurezza_server$ arp
Indirizzo TipoHW IndirizzoHW Flag Maschera Interfaccia
192.168.56.104           ether   08:00:27:e7:5a:5b   C                     enp0s3
192.168.56.100           ether   08:00:27:53:64:7d   C                     enp0s3
192.168.56.106           ether   08:00:27:e7:5a:5b   C                     enp0s3
```

Attacker:

```
kali@kali:~/Desktop/Exp6$ sudo arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.56.106           ether   08:00:27:f0:d6:29   C                     eth0
192.168.56.100           ether   08:00:27:53:64:7d   C                     eth0
192.168.56.109           ether   08:00:27:42:7a:21   C                     eth0
```

So thanks to our script it is possible, thought the right configuration of the filter, to alter a connection between two IP addresses. In our case we has limited the filter to, as said before, change all TCP packet data. Instead, following the other method, we should control all the sequence numbers and the acknowledge numbers, because if they don't match, the PC refuse that packet because it wasn't what is waiting for. In fact the sequence number follow a rule, it is the number of the next byte, so for each packet its new value is the old one plus the dimension of the packet data in byte.

Finally we tried to perform the attack and it works properly, as in the received file all the spaces are substituted by smiles.