ARP Cache Poisoning Attack Lab

1 Task 1: ARP Cache Poisoning

The objective of this task is to use packet spoofing to launch an ARP cache poisoning attack on a target, such that when two victim machines A and B try to communicate with each other, their packets will be intercepted by the attacker, who can make changes to the packets, and can thus become the man in the middle between A and B. This is called Man-In-The-Middle (MITM) attack. In this lab, we use ARP cahce poisoning to conduct an MITM attack.

The following code skeleton shows how to construct an ARP packet using Scapy.

```
#!/usr/bin/python
from scapy.all import *

E = Ether()
A = ARP()

pkt = E/A
sendp(pkt)
```

The above program constructs and sends an ARP packet. Please set necessary attribute names/values to define your own ARP packet. We can use `ls(ARP)` to see the attribute names of the ARP class. If a field is not set, a default value will be used (see the third column of the output):

```
>>> ls(ARP)
hwtype     : XShortField                          = (1)
ptype      : XShortEnumField                      = (2048)
hwlen      : ByteField                            = (6)
plen       : ByteField                            = (4)
op         : ShortEnumField                       = (1)
hwsrc      : ARPSourceMACField                    = (None)
psrc       : SourceIPField                        = (None)
hwdst      : MACField                             = ('00:00:00:00:00:00')
pdst       : IPField                              = ('0.0.0.0')
```

In this task, we have three VMs, A, B, and M. We would like to attack A's ARP cache, such that the following results is achieved in A's ARP cache.

```
B's IP address --> M's MAC address
```

There are many ways to conduct ARP cache poisoning attack. Students need to try the following three methods, and report whether each method works or not.

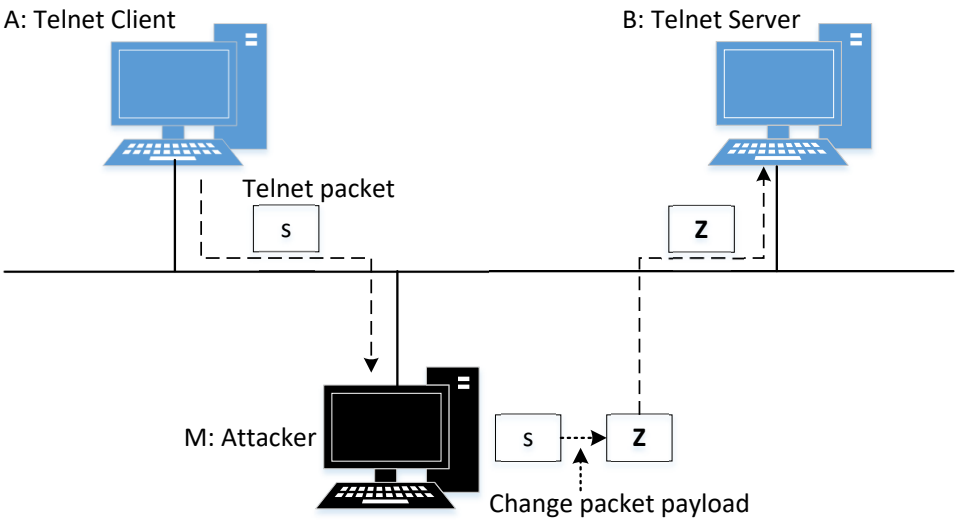

Figure 1: Man-In-The-Middle Attack against telnet

- **Task 1A (using ARP request).** On host M, construct an ARP request packet and send to host A. Check whether M's MAC address is mapped to B's IP address in A's ARP cache.
- **Task 1B (using ARP reply).** On host M, construct an ARP reply packet and send to host A. Check whether M's MAC address is mapped to B's IP address in A's ARP cache.
- **Task 1C (using ARP gratuitous message).** On host M, construct an ARP gratuitous packets. ARP gratuitous packet is a special ARP request packet. It is used when a host machine needs to update outdated information on all the other machine's ARP cache. The gratuitous ARP packet has the following characteristics:
  - The source and destination IP addresses are the same, and they are the IP address of the host issuing the gratuitous ARP.
  - The destination MAC addresses in both ARP header and Ethernet header are the broadcast MAC address (`ff:ff:ff:ff:ff:ff`).
  - No reply is expected.

2 Task 2: MITM Attack on Telnet using ARP Cache Poisoning

Hosts A and B are communicating using Telnet, and Host M wants to intercept their communication, so it can make changes to the data sent between A and B. The setup is depicted in Figure 1.

Step 1 (Launch the ARP cache poisoning attack). First, Host M conducts an ARP cache poisoning attack on both A and B, such that in A's ARP cache, B's IP address maps to M's MAC address, and in B's ARP cache, A's IP address also maps to M's MAC address. After this step, packets sent between A and B will all be sent to M. We will use the ARP cache poisoning attack from Task 1 to achieve this goal.

Step 2 (Testing). After the attack is successful, please try to ping each other between Hosts A and B, and report your observation. Please show Wireshark results in your report.

Step 3 (Turn on IP forwarding). Now we turn on the IP forwarding on Host M, so it will forward the packets between A and B. Please run the following command and repeat Step 2. Please describe your observation.

```
$ sudo sysctl net.ipv4.ip_forward=1
```

Step 4 (Launch the MITM attack). We are ready to make changes to the Telnet data between A and B. Assume that A is the Telnet client and B is the Telnet server. After A has connected to the Telnet server on B, for every key stroke typed in A's Telnet window, a TCP packet is generated and sent to B. We would like to intercept the TCP packet, and replace each typed character with a fixed character (say Z). This way, it does not matter what the user types on A, Telnet will always display Z.

From the previous steps, we are able to redirect the TCP packets to Host M, but instead of forwarding them, we would like to replace them with a spoofed packet. We will write a sniff-and-spoof program to accomplish this goal. In particular, we would like to do the following:

- We first keep the IP forwarding on, so we can successfully create a Telnet connection between A to B. Once the connection is established, we turn off the IP forwarding using the following command. Please type something on A's Telnet window, and report your observation:

  ```
  $ sudo sysctl net.ipv4.ip_forward=0
  ```

- We run our sniff-and-spoof program on Host M, such that for the captured packets sent from A to B, we spoof a packet but with TCP different data. For packets from B to A (Telnet response), we do not make any change, so the spoofed packet is exactly the same as the original one.

A skeleton sniff-and-spoof program is shown below:

```
#!/usr/bin/python
from scapy.all import *

def spoof_pkt(pkt):
   print("Original Packet.........")
   print("Source IP : ", pkt[IP].src)
   print("Destination IP :", pkt[IP].dst)

   a = IP()
   b = TCP()
   data = pkt[TCP].payload
   newpkt = a/b/data

   print("Spoofed Packet.........")
   print("Source IP : ", newpkt[IP].src)
   print("Destination IP :", newpkt[IP].dst)
   send(newpkt)

pkt = sniff(filter='tcp',prn=print_pkt)
```

The above program sniffs all the TCP packets and then spoof a new TCP packet based on the captured packets. Please make necessary changes to distinguish whether a packet is sent from A or B. If it is sent from A, set all the attribute names/values of the new packet to be the same as those of the original packet,

and replace each alphanumeric characters in the payload (usually just one character in each packet) with character Z. If the captured packet is sent from B, no change will be made.

In Telnet, every character we type in the Telnet window will trigger a TCP packet. Therefore, in a typical Telnet packet from client to server, the payload only contains one character. The character will then be echoed back by the server, and the client will then display the character in its window. Therefore, what we see in the client window is not the direct result of the typing; whatever we type in the client window takes a round trip before it is displayed. If the network is disconnected, whatever we typed on the client window will not displayed, until the network is recovered. Similarly, if attackers change the character to Z during the round trip, Z will be displayed at the Telnet client window.

Here is a summary what we need to do in order to launch the MITM attack.

- Conduct ARP cache poisoning attacks against Hosts A and B.
- Turn on IP forwarding on Host M.
- Telnet from host A to Host B.
- After the Telnet connection has been established, turn off IP forwarding.
- Conduct the sniff and spoof attack on Host M.

3 Submission

Students need to submit a detailed lab report to describe what they have done, what they have observed, and how they interpret the results. Reports should include evidences to support the observations. Evidences include packet traces, screenshots, etc. Reports should also list the important code snippets with explanations. Simply attaching code without any explanation will not receive credits.