Cmpt 281 - Lab 3
Darion Kwasnitza - 3122890

Part 1 - Task1.1

Set up Lan using docker-compose up:

```
[02/01/24]seed@VM:~/.../Labsetup$ docker-compose up
WARNING: Found orphan containers (hostB-10.9.0.6, seed-attacker, hostA-10.9.0.5)
 for this project. If you removed or renamed this service in your compose file,
you can run this command with the --remove-orphans flag to clean it up.
Starting M-10.9.0.105 ... done
Starting B-10.9.0.6   ... done
Starting A-10.9.0.5   ... done
Attaching to A-10.9.0.5, B-10.9.0.6, M-10.9.0.105
B-10.9.0.6 |  * Starting internet superserver inetd                    [ OK ]
A-10.9.0.5 |  * Starting internet superserver inetd                    [ OK ]
```

Find attacker container ID:

```
[02/01/24]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID        IMAGE                                  COMMAND
 CREATED               STATUS              PORTS               NAMES
e52aba361133        handsonsecurity/seed-ubuntu:large   "bash -c ' /etc/init…"
 About a minute ago   Up About a minute                       B-10.9.0.6
3c310573897e        handsonsecurity/seed-ubuntu:large   "/bin/sh -c /bin/bash"
 About a minute ago   Up About a minute                       M-10.9.0.105
aa404aef3d26        handsonsecurity/seed-ubuntu:large   "bash -c ' /etc/init…"
 About a minute ago   Up About a minute                       A-10.9.0.5
[02/01/24]seed@VM:~/.../Labsetup$
```

Execute /bin/bash commands for M and A:

```
[02/01/24]seed@VM:~/.../Labsetup$ docker exec -it 3c310573897e /bin/bash
root@3c310573897e:/#
```

```
[02/01/24]seed@VM:~/.../Labsetup$ docker exec -it aa404aef3d26 /bin/bash
root@aa404aef3d26:/#
```

Check ARP table on host A and set the ARP table by pinging 10.9.0.6:

```
root@aa404aef3d26:/# arp -n
root@aa404aef3d26:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.141 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.100 ms
^C
--- 10.9.0.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1014ms
rtt min/avg/max/mdev = 0.100/0.120/0.141/0.020 ms
root@aa404aef3d26:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@aa404aef3d26:/#
```

Start attack from M using ./volumes/arp_request.py:

```
root@3c310573897e:/# ./volumes/arp_request.py
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
root@3c310573897e:/#
```

Check the ARP table on host A again; now 10.9.0.99 is showing with the address aa:bb:cc:dd:ee:ff

```
root@aa404aef3d26:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.99                ether   aa:bb:cc:dd:ee:ff   C                     eth0
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@aa404aef3d26:/#
```

Part 1 - Task1.2

Start the attack using ./volumes/arp_reply.py:

```
root@3c310573897e:/# ./volumes/arp_reply.py
SENDING SPOOFED ARP REPLY......
.
Sent 1 packets.
root@3c310573897e:/#
```

Recheck the ARP table and notice the difference in the address of 10.9.0.99: It is now aa:bb:cc:dd:00:11, which was not the original; the reply is actually different from the request.

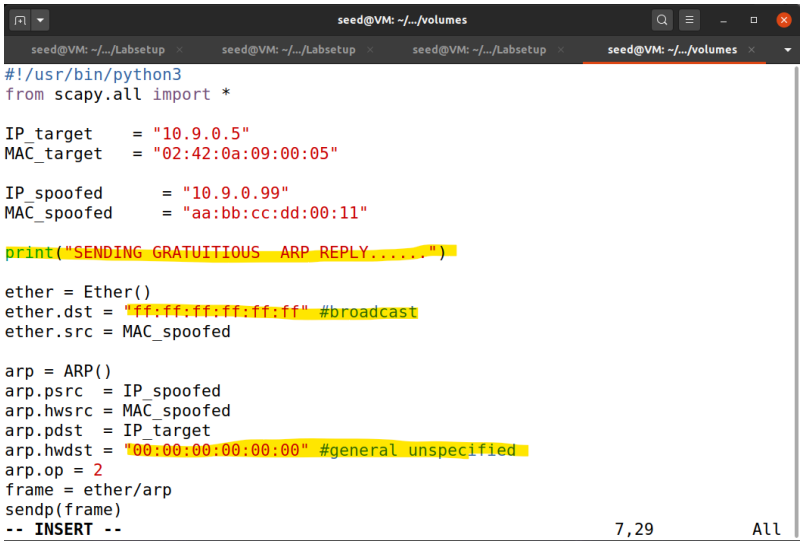
```
root@aa404aef3d26:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.99                ether   aa:bb:cc:dd:ee:ff   C                     eth0
10.9.0.6                 ether   aa:bb:cc:dd:00:11   C                     eth0
root@aa404aef3d26:/#
```

```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.99                ether   aa:bb:cc:dd:00:11   C                     eth0
10.9.0.6                 ether   aa:bb:cc:dd:00:11   C                     eth0
root@aa404aef3d26:/#
```

I tried to do it with 10.9.0.98; however, the attack failed because there was no cached entry for the target IP address (Note: switched order of 10.9.0.98 and 10.9.0.99).

Part 2:

I made the .dst of the ether to be a broadcast channel to all addresses and then made the hwdst to be all as well by using a general unspecified address:



This gave me the following result:

```
root@aa404aef3d26:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.1                 ether   aa:bb:cc:dd:00:11   C                     eth0
10.9.0.99                ether   aa:bb:cc:dd:00:11   C                     eth0
10.9.0.6                 ether   aa:bb:cc:dd:00:11   C                     eth0
root@aa404aef3d26:/#
```

This attack changed all IP addresses to the same HWaddress, which is what a gratuitous ARP cache attack does; it targets all of the LAN members' ARP cache.