

Progress report (Iteration I) Results

Vaskovskaya Daria

February 2020

1 Goal

Study different variants of simulation platforms and set up the environment for Man In The Middle Attacks, perform some MITM attacks on simulation platforms

2 Choice of simulation platform

To perform MITM attacks and test ways to defend from them there should exist the environment to test on. For this purpose the network emulators were studied.

There are many both free/open-source and proprietary network simulators available. The choice was made out of Open-Source Network Simulators.

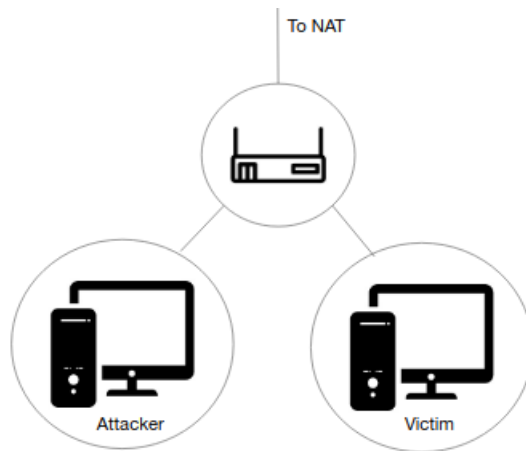
The criteria for the choice were good documentation, simplicity and flexibility.

The first choice fell on the Virtual Box [1] for its flexibility, my experience with it, and community support.

However, there are also out-of-the-box solutions. After reading documentations and launching a few of them I've chosen mininet [2] as one of the best network simulators for the good documentation and simplicity. It is also controlled by python scripts and I have some experience with python.

There would be presented a way to emulate the network in VirtualBox. And, if mininet will be capable of all the required configurations for the showcase at the end of a project, I will use mininet at the presentation because of an easier set-up.

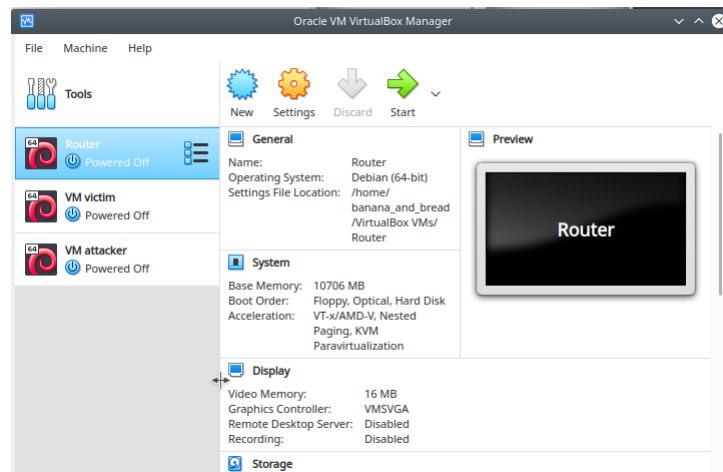
Network topology will look like this:



3 Virtual Box set-up

3.1 Virtual Machines and their settings

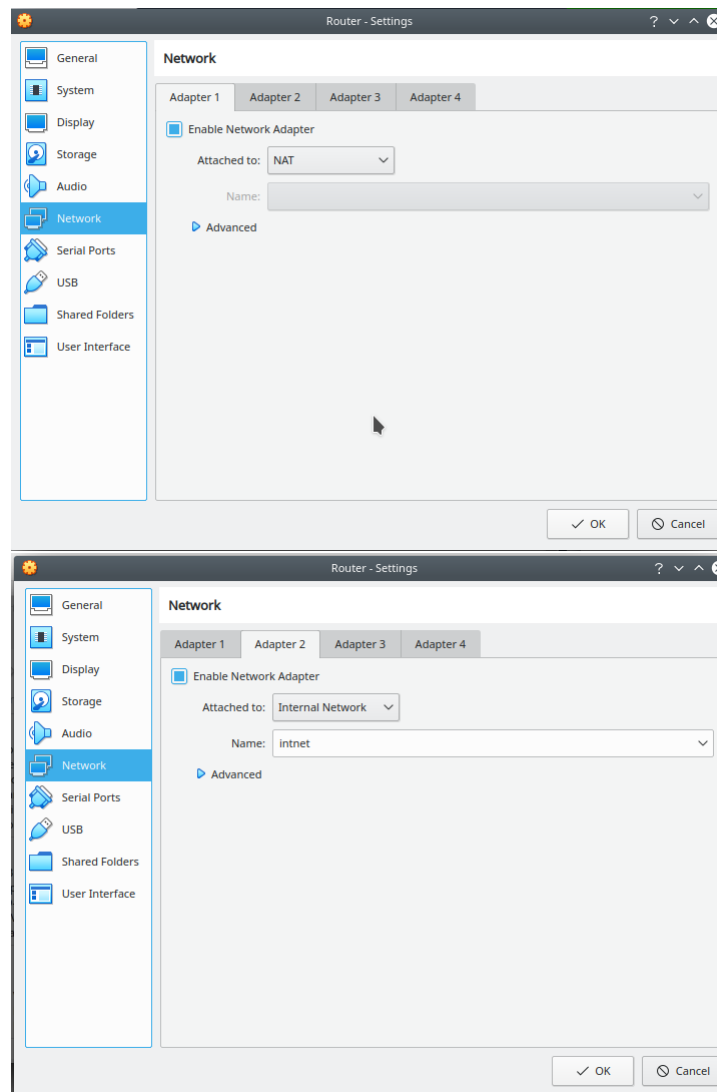
Three machines were created:



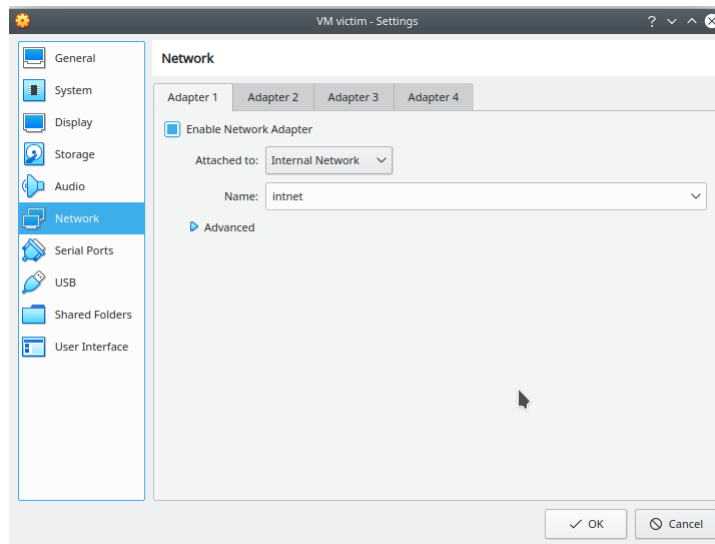
OS: linux-lite-4.6

Network settings are the following:

Router:



Other machines:



3.2 Router set up

Script for the set up is available on github.com [4]

3.2.1 Quagga

Quagga [6] is a network routing software suite providing implementations of Open Shortest Path First, Routing Information Protocol, Border Gateway Protocol and IS-IS for Unix-like platforms, particularly Linux, Solaris, FreeBSD and NetBSD.

Step-by step setup:

```
sudo apt-get update
sudo apt install quagga
sudo apt install quagga-doc
sed -i '/^#.*net.ipv4.ip_forward=1/s/^#//' /etc/sysctl.conf
cp /usr/share/doc/quagga-core/examples/vtysh.conf.sample\
/etc/quagga/vtysh.conf
cp /usr/share/doc/quagga-core/examples/zebra.conf.sample\
/etc/quagga/zebra.conf
cp /usr/share/doc/quagga-core/examples/bgpd.conf.sample\
/etc/quagga/bgpd.conf
sudo chown quagga:quagga /etc/quagga/*.conf
sudo chown quagga:quaggavty /etc/quagga/vtysh.conf
sudo chmod 640 /etc/quagga/*.conf
```

```
sudo service zebra start
```

```
sudo service bgpd start
```

```
sudo systemctl is-enabled zebra.service
sudo systemctl is-enabled bgpd.service
sudo systemctl enable zebra.service
sudo systemctl enable bgpd.service
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
ufw disable
```

3.2.2 DHCP

Run the the following commands to enable DHCP: :

```
sudo apt install isc-dhcp-server
```

```
echo 'default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
authoritative;
subnet 192.168.50.0 netmask 255.255.255.0 {
range 192.168.50.50 192.168.50.100;
option routers 192.168.50.1;
option subnet-mask 255.255.255.0;
option domain-name-servers 192.168.50.1, 8.8.8.8;
}' > /etc/dhcp/dhcpd.conf
sudo systemctl restart isc-dhcp-serv
```

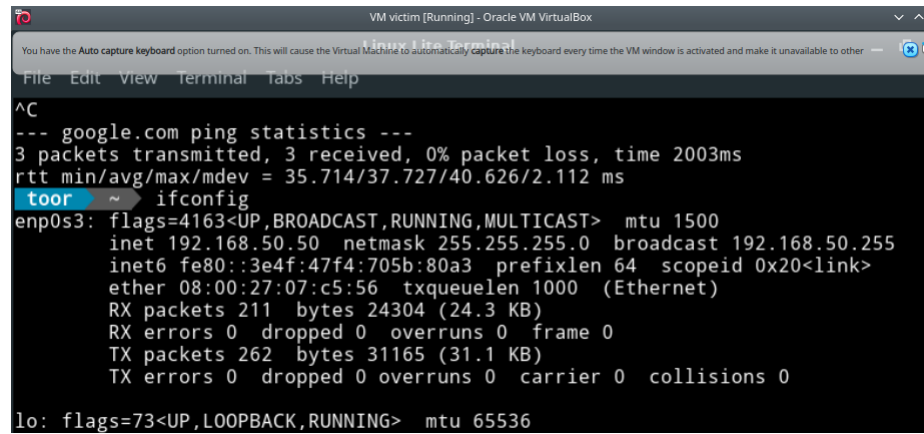
```
echo '
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
address 192.168.50.1
netmask 255.255.255.0
dns-nameservers 8.8.8.8
' > /etc/network/interfaces
```

```
sudo ip a flush enp0s8
sudo systemctl restart networking.service
sudo systemctl restart isc-dhcp-server
```

3.3 Tests

DHCP gave address to a victim machine:

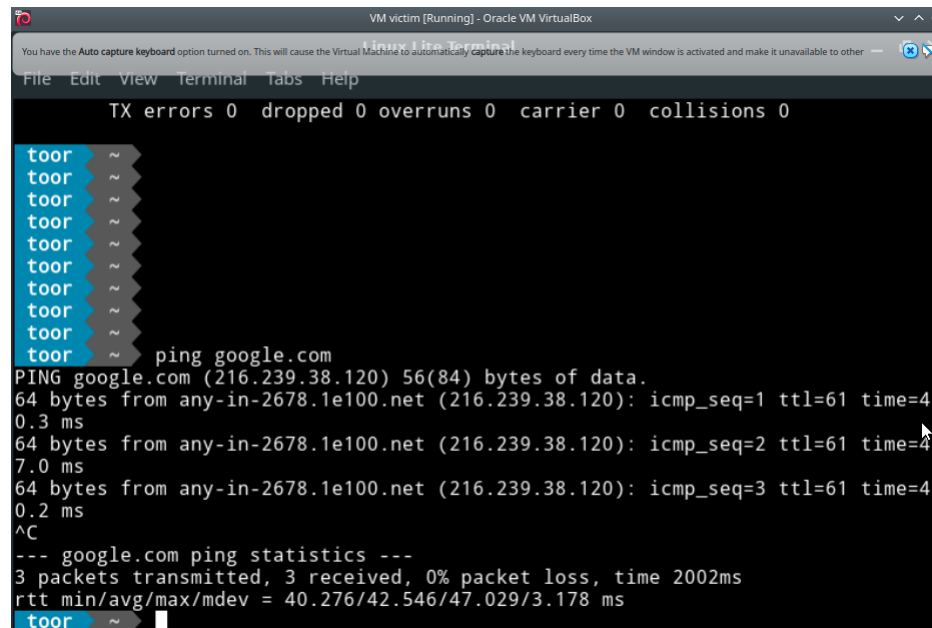


The screenshot shows a terminal window titled "VM victim [Running] - Oracle VM VirtualBox". The terminal output displays the results of the 'ifconfig' command for the 'enp0s3' interface. It shows the interface is up and running with an IP address of 192.168.50.50, a netmask of 255.255.255.0, and a broadcast address of 192.168.50.255. It also shows the MAC address 08:00:27:07:c5:56 and various statistics for RX and TX packets and errors.

```
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 35.714/37.727/40.626/2.112 ms
toor ~ # ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.50.50 netmask 255.255.255.0 broadcast 192.168.50.255
        inet6 fe80::3e4f:47f4:705b:80a3 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:07:c5:56 txqueuelen 1000 (Ethernet)
        RX packets 211 bytes 24304 (24.3 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 262 bytes 31165 (31.1 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

Internet became available via router:



The screenshot shows a terminal window titled "VM victim [Running] - Oracle VM VirtualBox". The terminal output shows the results of a 'ping google.com' command. It displays the IP address of google.com (216.239.38.120) and the results of four ping attempts, showing 0% packet loss and a time of 4.0 ms. It also shows the results of the 'ping statistics' command, which shows 3 packets transmitted, 3 received, 0% packet loss, and a time of 2002ms.

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

toor ~ #
toor ~ #
toor ~ #
toor ~ #
toor ~ #
toor ~ #
toor ~ #
toor ~ #
toor ~ # ping google.com
PING google.com (216.239.38.120) 56(84) bytes of data:
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=1 ttl=61 time=4
0.3 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=2 ttl=61 time=4
7.0 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=3 ttl=61 time=4
0.2 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 40.276/42.546/47.029/3.178 ms
toor ~ #
```

4 MITM framework

4.1 Choice of MITM framework

I have chosen bettercap [3], mostly because it is supported in 2020. Script for the set up is available on github.com [5]

4.2 Some attacks showcase

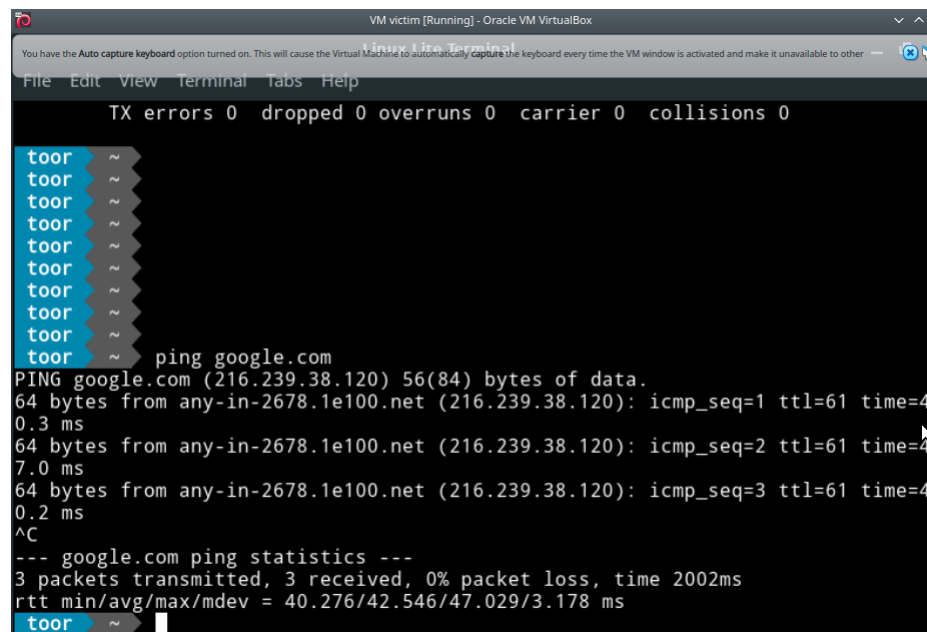
4.2.1 ARP spoofing

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. It is possible to make ARP spoofing attacks with Bettercap via:

```
set arp.spoof.targets 192.168.50.1/24
```

```
arp.spoof on
```

Victim's request is seen on the attacker machine:

A screenshot of a VirtualBox terminal window titled "VM victim [Running] - Oracle VM VirtualBox". The terminal shows network statistics at the top: "TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0". Below this, there is a list of network packets, each starting with "toor" and a tilde "~". The terminal then shows the command "ping google.com" being executed. The output of the ping command is displayed, showing three successful pings to google.com (216.239.38.120) with 56(84) bytes of data. The ping statistics at the bottom show "3 packets transmitted, 3 received, 0% packet loss, time 2002ms" and "rtt min/avg/max/mdev = 40.276/42.546/47.029/3.178 ms".

```
VM victim [Running] - Oracle VM VirtualBox
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other...
File Edit View Terminal Tabs Help

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

toor ~
toor ~
toor ~
toor ~
toor ~
toor ~
toor ~
toor ~
toor ~
toor ~ ping google.com
PING google.com (216.239.38.120) 56(84) bytes of data.
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=1 ttl=61 time=4
0.3 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=2 ttl=61 time=4
7.0 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=3 ttl=61 time=4
0.2 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 40.276/42.546/47.029/3.178 ms
toor ~
```

References

- [1] VirtualBox
<https://www.virtualbox.org/>

- [2] mininet
<http://mininet.org/>
- [3] bettercap
<https://www.bettercap.org/>
- [4] Script for setting up router
https://github.com/BananaAndBread/Project/blob/master/scripts/router_script.sh
- [5] Script for setting up bettercap
https://github.com/BananaAndBread/Project/blob/master/scripts/bettercap_script.sh
- [6] Quagga
<https://www.quagga.net/>