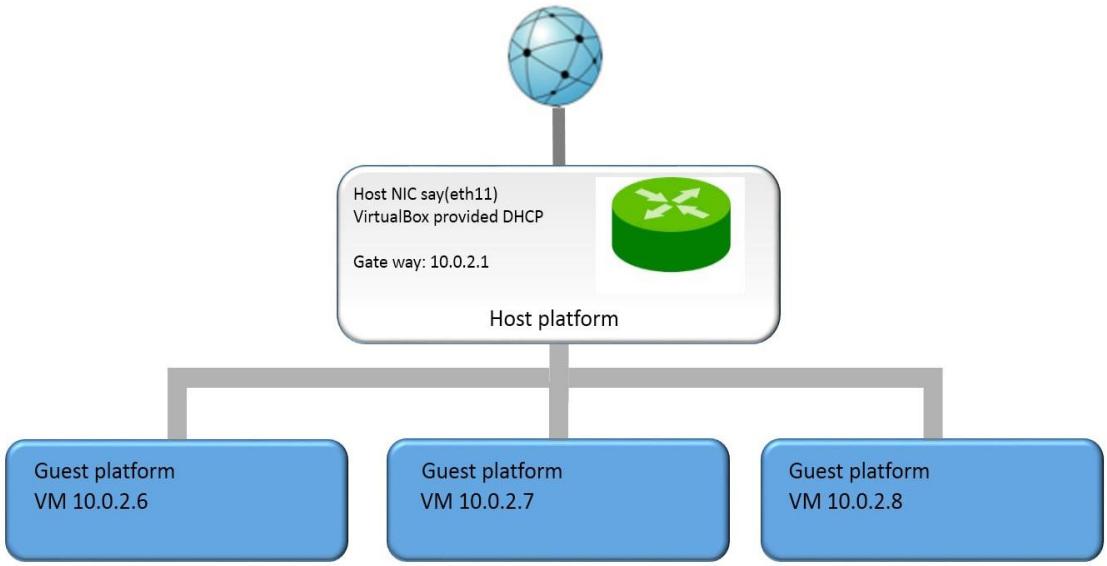
Project Manual

CS- 458

Network Configuration in VirtualBox for the Project

In many of the labs, we need to run multiple guest VMs, and these VMs should be able to (1) reach out to the Internet, (2) communicate with each other. In the VirtualBox, if we use the "NAT" setting (default setting) for each VM, we can achieve 1, but not 2, because each VM will be placed in its own private network, not on a common one; they even have the same IP address, which is not a problem because each VM is the only computer on its own private network. On the other hand, if we use the "Host-only" setting for each VM, we can achieve 2, but not 1. Using this setting, all the VMs and the host will be put on a common network, so they can communicate with each other; however, due to the lack of NAT, the VMs cannot reach

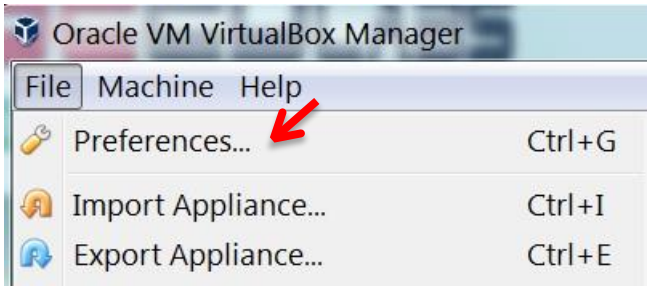


out to the outside.

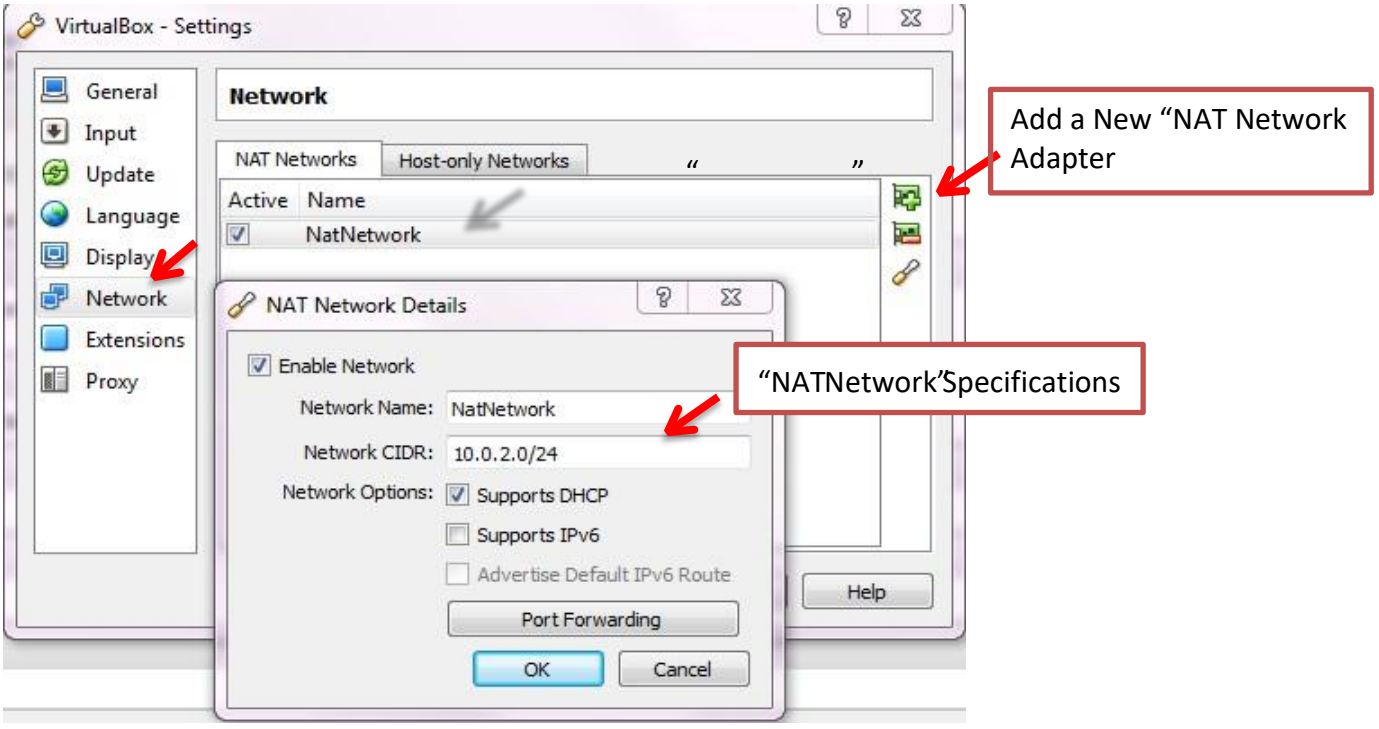
Therefore, in order to achieve all these 2 goals, we have to use a network adapter called **"NAT Network"**. The adapter works in a similar way to "local area network" or LAN. It enable VMs communication within same local network as well as the communication to the internet. All the communication goes through this single adapter. As show in Figure 1, gateway router transfers the packets among the VMs and transfers the packets from local network to Internet.

Configuration Instruction

Step 1: Make sure you are using the most up-to-date VirtualBox. As show in the following figure, click the “File” on the top left of the VirtualBox main UI. Then choose “Preferences…” option.

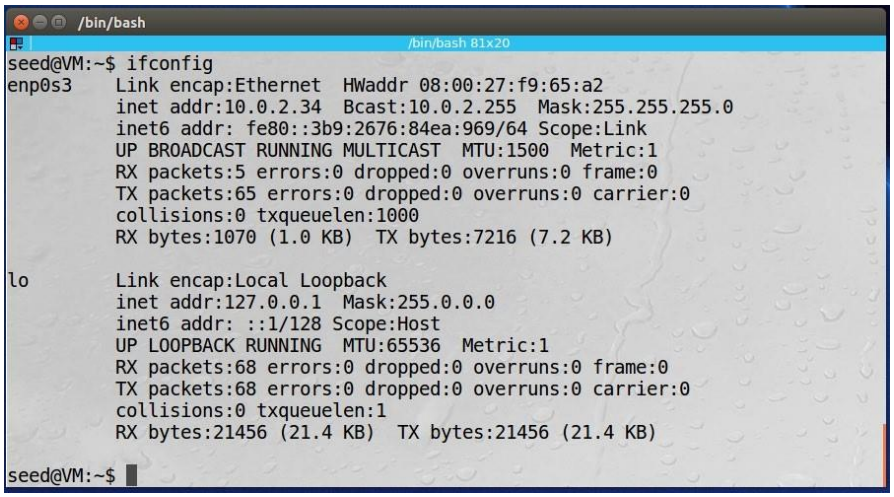


Step 2: Click the “Network” tab on left panel. click the “+” button to create a new NAT Networks (NatNetwork) adaptor (if one does not exist). Double click on the NatNetwork and look at its specifications. Set the specifications as the same as what is shown below.



Step 4: Go to VM setting, you need to power off the VM before making the following changes. Enable Adapter 1(at the same time, disable the other adapters), and choose “NAT Network”.

Step 5: Now power on the VMs and check the IP address.



Troubleshooting:

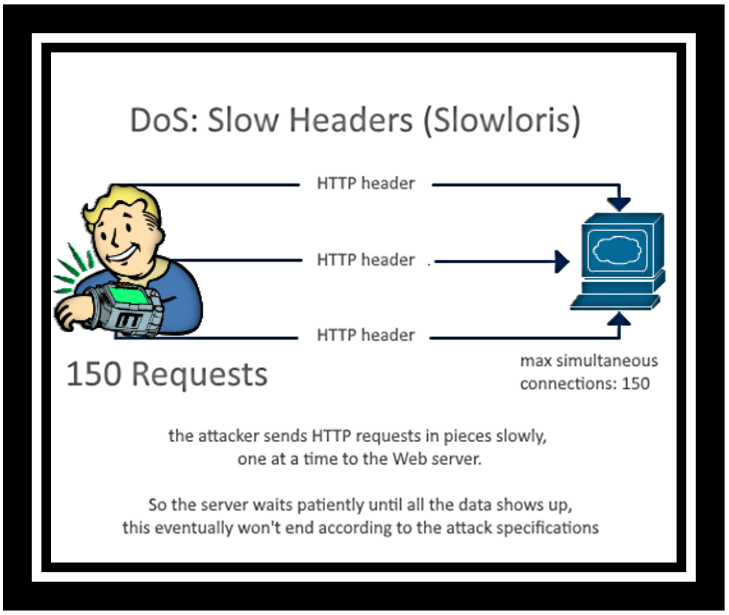
- If VMs cannot ping each other, refreshing the MAC Address can resolve the issue.

Instructions to generate attack:

Disclaimer: This tool should only be used for testing the resiliency of the VM (server) against DoS and DDoS attacks. Neither the instructor nor the university are responsible for any misuse of these tools. These tools are to be used only in the virtual environment.

"Slowloris" or "Slow HTTP"

Most of web administrators don't care properly about the security of the servers, are often target of attacks that a lot of black hat hackers know how to perform in mass. One of those



tricky attacks are the Slow HTTP attacks that target any kind of web server. Figure explains how the attack looks like.

Running the attack

Slowloris and Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. This tool is sending partial HTTP requests, trying to get denial of service from target HTTP server.

$ slowhttptest -c 500 -H -g -o ./output_file -i 10 -r 200 -t GET -u http://yourwebsite-or-server-ip.com -x 24 -p 2

-c: Specifies the target number of connections to establish during the test (in this example 500, normally with 200 should be enough to hang a server that doesn't have protection against this attack).

-H: Starts slowhttptest in SlowLoris mode, sending unfinished HTTP requests.

-g: Forces slowhttptest to generate CSV and HTML files when test finishes with timestamp in filename.

-o: Specifies custom file name, effective with -g.

-i: Specifies the interval between follow up data for slowrois and Slow POST tests (in seconds).

-r: Specifies the connection rate (per second).

-t: Specifies the verb to use in HTTP request (POST, GET etc).

-u: Specifies the URL or IP of the server that you want to attack.

-x: Starts slowhttptest in Slow Read mode, reading HTTP responses slowly.

-p: Specifies the interval to wait for HTTP response onprobe connection, before marking the server as DoSed (in seconds).

Capturing Data from the victim VM (server)

`tcpdump` (packet capture tool)

Man Page: https://www.tcpdump.org/manpages/tcpdump.1.html

`tcpdump` is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

Simple command to capture all the incoming and outgoing packets:

```
$ tcpdump -i enp0s3 -w file.pcap -vv
```

```
-i: interface

-w: write output to a file; When the -w option is used; the output
is not displayed on the screen.

-vv: for full protocol decode
```

`vmstat` - Virtual memory statistics**:**

Man page: https://linux.die.net/man/8/vmstat

`vmstat` reports information about processes, memory, paging, block IO, traps, and cpu activity.

`ss` - socket statistics

Man page: https://man7.org/linux/man-pages/man8/ss.8.html

The **`ss`** command is a tool used to dump socket statistics and displays information in similar fashion (although simpler and faster) to netstat. The ss command can also display even more TCP and state information than most other tools.

`mpstat` - processors related statistics

Man page: https://linux.die.net/man/1/mpstat

The **`mpstat`** command writes to standard output activities for each available processor, processor 0 being the first one. Global average activities among all processors are also reported. If no activity has been selected, then the default report is the CPU utilization report.