Practicum of Attack and Defense of Network Security

Lab01: VMs Setup and Network Config

Ref: GTW

Learning Goals

- Setup and clone Virtual Machines (VM) using Oracle VirtualBox and the pre-built SEED Ubuntu image.
- Configure VMs with different network settings.
- Get familiar with basic Linux network tools, such as "ping" and "traceroute"

Tasks

Task 1: Install Oracle VirtualBox

Task 2: Install VMs using Pre-Built SEED Images

Task 3: Setup shared folder between host and VMs

Task 4: Test Communication between VMs on the same host

Task 5: Test Communication between *bridged* VMs on Different Hosts

Task 6: Reconstruct ARP cache by iteratively PING all subnet IP addresses

Task 7: Show path to remote machines by TRACEROUTE

Task 8: Test Communication between VMs on Different Hosts using NAT

Lab Procedure

Task 1: Install Oracle VirtualBox

Step 1: From the Oracle VirtualBox download site, download the latest Version for your OS.

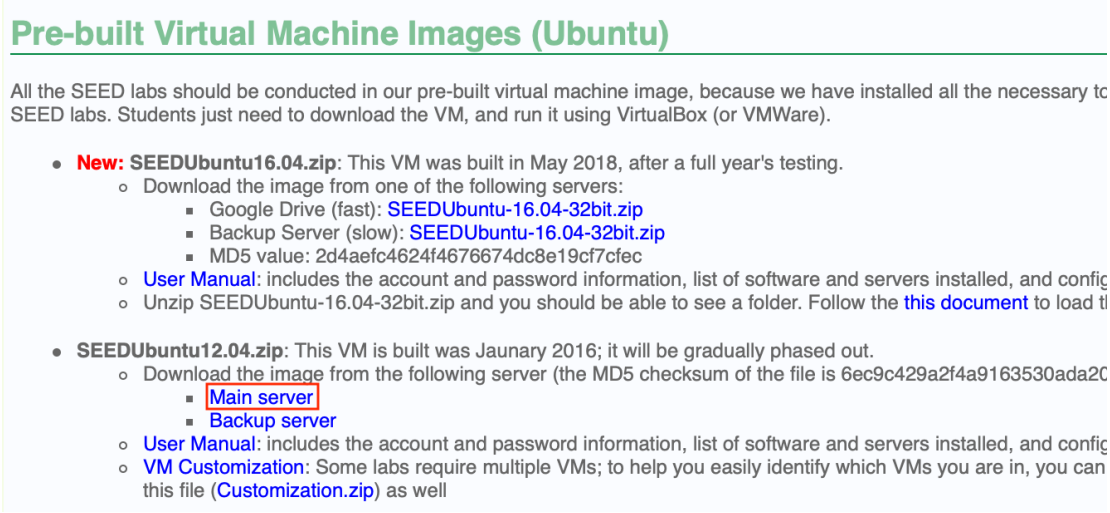
http://www.virtualbox.org/wiki/Downloads

Step 2: Install VirtualBox

- Follow the default setting until installation finished.
- Launch VirtualBox

Task 2: Install VMs using Pre-Built SEED Images

Step 1: From the SEED Project web site (http://www.cis.syr.edu/~wedu/seed/lab_env.html), download the Ubuntu images (SEEDUbuntu1204.zip). It is a ZIP file, so extract the files. Confirm the existence of a VMDK file.



Pre-built Virtual Machine Images (Ubuntu)

All the SEED labs should be conducted in our pre-built virtual machine image, because we have installed all the necessary to SEED labs. Students just need to download the VM, and run it using VirtualBox (or VMWare).

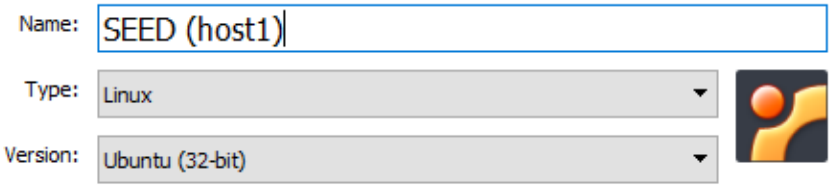
- **New: SEEDUbuntu16.04.zip**: This VM was built in May 2018, after a full year's testing.
  - Download the image from one of the following servers:
    - Google Drive (fast): SEEDUbuntu-16.04-32bit.zip
    - Backup Server (slow): SEEDUbuntu-16.04-32bit.zip
    - MD5 value: 2d4aefc4624f4676674dc8e19cf7cfec
  - User Manual: includes the account and password information, list of software and servers installed, and config
  - Unzip SEEDUbuntu-16.04-32bit.zip and you should be able to see a folder. Follow the this document to load t
- **SEEDUbuntu12.04.zip**: This VM is built was Jaunary 2016; it will be gradually phased out.
  - Download the image from the following server (the MD5 checksum of the file is 6ec9c429a2f4a9163530ada20
    - Main server
    - Backup server
  - User Manual: includes the account and password information, list of software and servers installed, and config
  - VM Customization: Some labs require multiple VMs; to help you easily identify which VMs you are in, you can this file (Customization.zip) as well

(Note that we are using the older version of SEED because the newer version is not fully tested yet.)

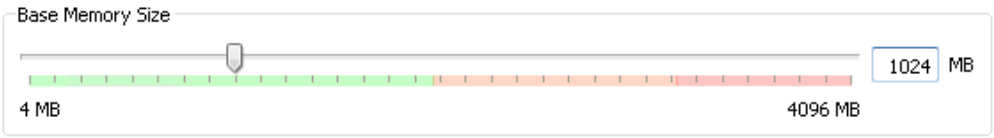
Step 2: Start VirtualBox and then select 

to create a new virtual machine (VM).

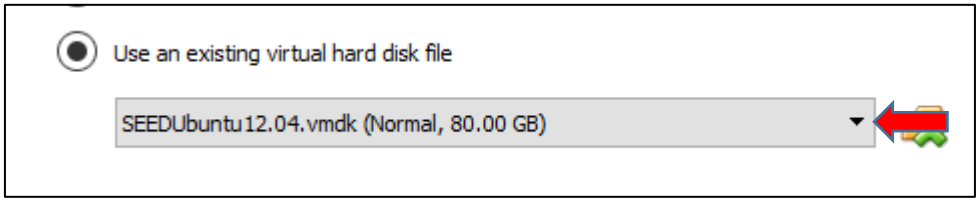
Step 3: Name the host and select its OS



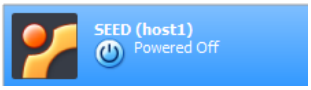
Step 4: Allocate RAM. The SEED project recommends 512M, and you should allocate more if your computer has large memory.



Step 5: Select the VMDK file from the previous downloaded SEED image.

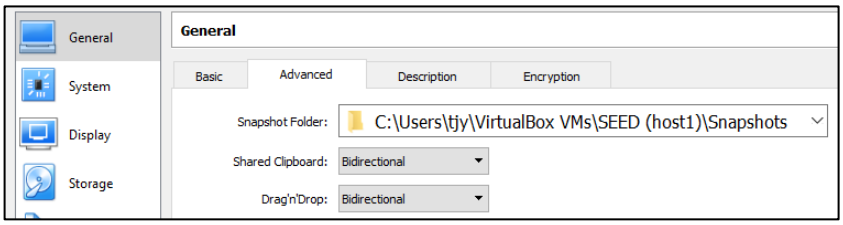


A new VM is created after this step:

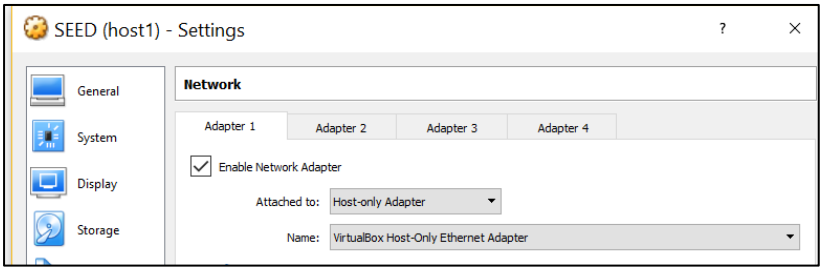


Step 6: Configure the VM (Note that these settings can only be modified when the VM is Powered-Off)

- General setting (advanced tab)



- Network setting (Host-only Adapter)



- If you encounter any problem regarding to "System Acceleration Settings" and cannot save changes, you have 2 choices to fix the problem:
  (a) Go to "Settings -> System -> Acceleration" and untick "hardware virtualization"
  (b) Enable Intel VT or AMD-V from BIOS. Please refer to this document.

Step 7: Start the new VM by double-click it. Then login with default accounts:

- User ID: ***root***; Password: ***seedubuntu***
- User ID: ***seed***; Password: ***dees***

Step 8: Start a new terminal session by clicking the terminal icon:

Step 9: Change the password for "seed"

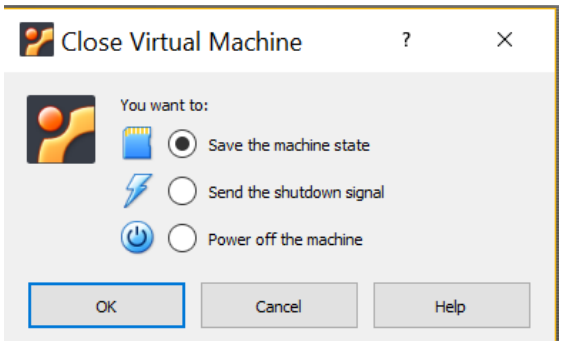
```
[03/29/2018 16:49] seed@ubuntu:~$ passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
[03/29/2018 16:49] seed@ubuntu:~$ _
```

Step 10: Check the IP address of your new VM:

```
[VM1] ifconfig
eth13     Link encap:Ethernet  HWaddr 08:00:27:cf:0b:c6
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fecf:bc6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KB)  TX bytes:8617 (8.6 KB)
```

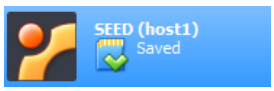
Step 11: "Save" or "Power off" the VM

- "Save" will save all configuration data and it will be much quicker to restart the VM.
- "Power off" will erase some states and you may need to reconfigure them later.
- In general, you should "Save" instead of "Power off" a VM.
- In the case of cloning a VM, the VM being cloned should be "Powered off."



After clicking "Save", the state of the VM is saved and it is also turned off.

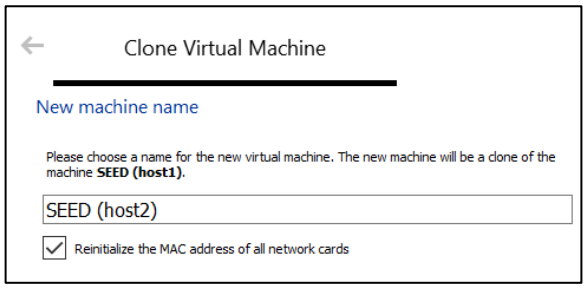
Click the VM to restart it and it will restore to its original state.



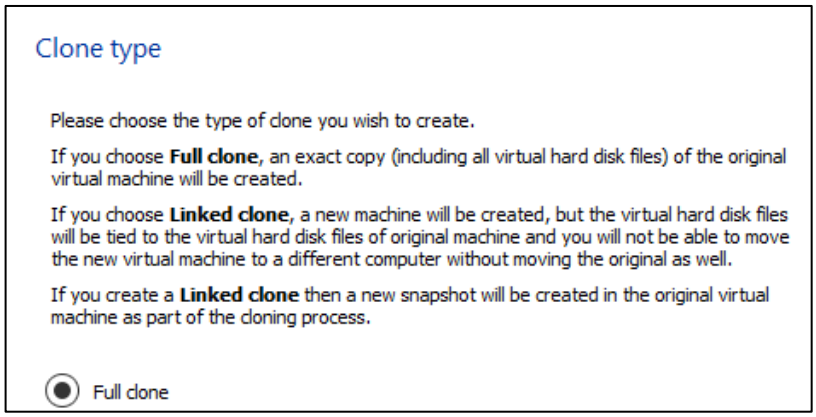
Now we are going to clone the VM.

Step 12: Make sure the VM being cloned is powered off.

Step 13: Right click the VM and select “Clone”. The clone dialog box will pop up. Name the clone VM host2.



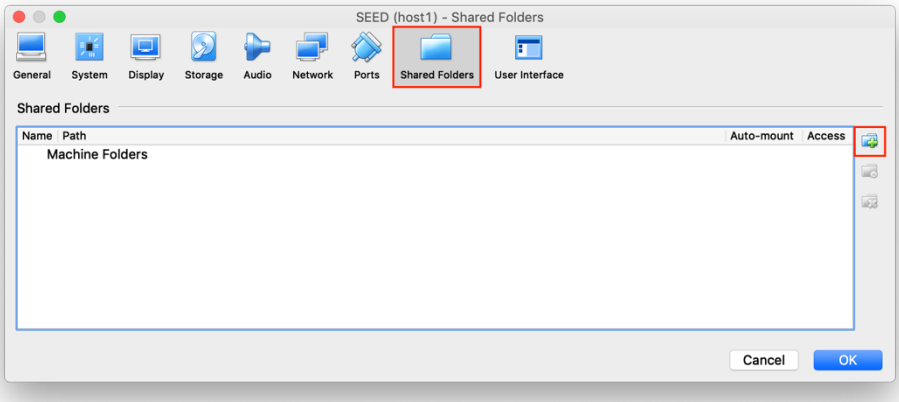
Step 14: Choose “Full clone”



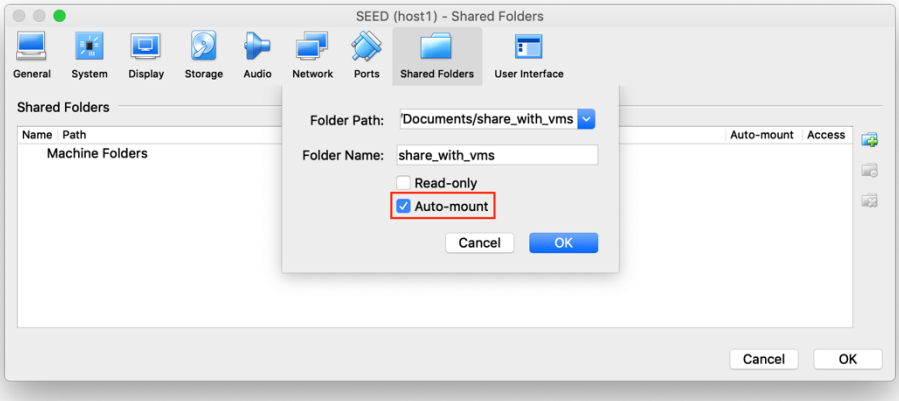
Step 15: Try to launch the newly-cloned VM.

Task 3: Setup shared folder between host and VMs

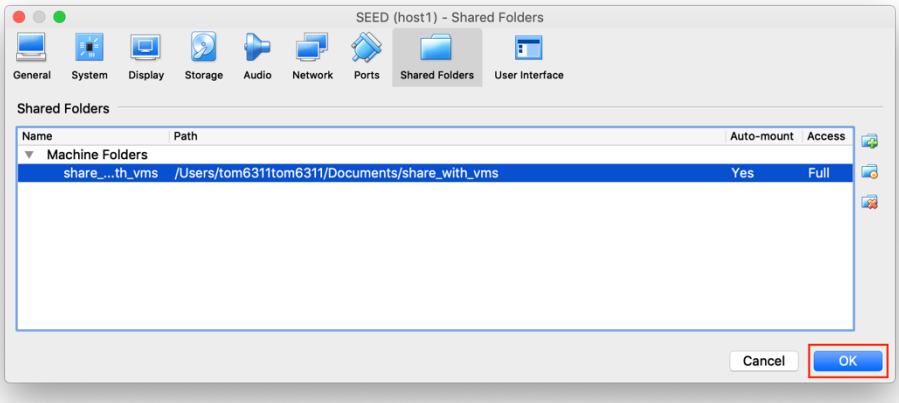
Step 1: From VirtualBox, power off VM1 and then right-click on it -> “Settings” -> “Shared Folders” -> “Add Folder”



Step 2: Select a folder path on your host to share with VMs. Check “Auto-mount”. Click “OK”



Step 3: Click “OK”. Then start VM1.



Step 4: On VM1, edit "/etc/group".

```
[VM1] sudo vi /etc/group
```

Step 5: In "/etc/group", find the group name "vboxsf" and add account "seed" into the group by typing "*:seed*" after the group ID. Then save the file.

```
vboxsf:x:1001:seed
```

Step 6: Logout and login again.

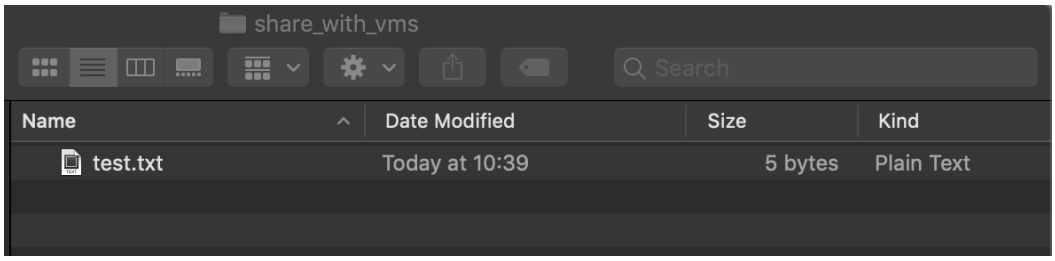
Step 7: Now you have permission to access the shared folder from the VM. It is mounted under "/media/". Go check for it.

```
[VM1] cd /media/sf_share_with_vms/
```

Step 8: Try to create a file in the shared folder

```
[VM1] echo 'test' > test.txt
```

Step 9: On your host (computer), go to the shared path and verify if "test.txt" is in it.

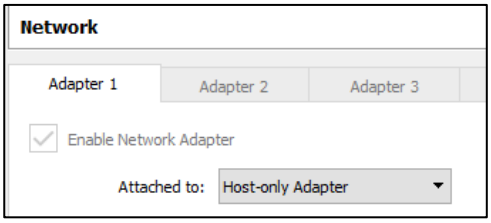


Step 10: Try to edit "test.txt" from the host and check if changes is applied on the VM.

Task 4: Test Communication between VMs on the same host

Figure 1. VM Communication on the same host.

Step 1: Choose Host-Only adapter for VM1 and VM2



Step 2: Verify the IP addresses on VM1 and VM2.

Screenshot 1

```
[VM1] ifconfig eth13
eth13     Link encap:Ethernet  HWaddr 08:00:27:cf:0b:c6
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fecf:bc6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:67 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13505 (13.5 KB)  TX bytes:10312 (10.3 KB)
```

Screenshot 2

```
[VM2] ifconfig
eth14     Link encap:Ethernet  HWaddr 08:00:27:c2:da:3e
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec2:da3e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1504 (1.5 KB)  TX bytes:10521 (10.5 KB)
```

Step 3: Check the connectivity between VM1 and VM2

Screenshot 3

```
[VM1] sudo ping -f -c 100 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.

--- 192.168.56.102 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 56ms
rtt min/avg/max/mdev = 0.148/0.273/0.759/0.107 ms, ipg/ewma 0.573/0.403 ms
```

Task 5: Test Communication between bridged VMs on Different Hosts

Task 5 requires two physical computers (hosts) as illustrated in Figure 2. Note that when you change the network setting, you need to power off the VM.

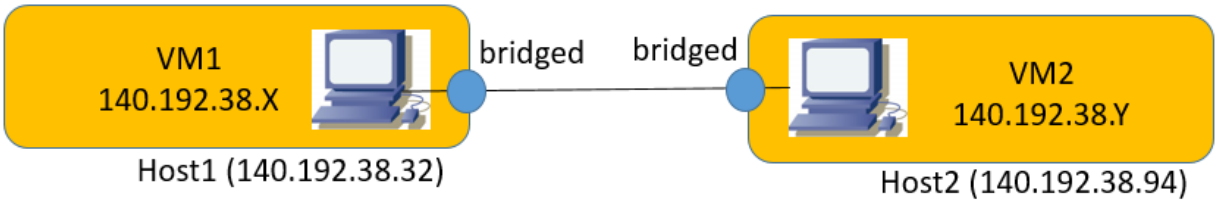
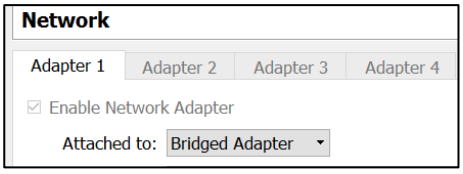


Figure 2. VM Communication between Bridged Hosts

Step 1: Provision a VM on each physical host and choose Bridged Adapter for its networking setting.



Step 2: The IP address of VM should be on the same subnet as the host. Verify it.

```
[VM1] ifconfig eth13
eth13     Link encap:Ethernet  HWaddr 08:00:27:a2:89:62
          inet addr:140.192.38.37  Bcast:140.192.38.255  Mask:255.255.255.0
```

```
[VM2] ifconfig eth14
eth14     Link encap:Ethernet  HWaddr 08:00:27:47:bf:da
          inet addr:140.192.38.46  Bcast:140.192.38.255  Mask:255.255.255.0
```

Step 3: Verify the connectivity between VM1 and VM2.

Screenshot 4

```
[VM1] sudo ping -f -c 100 140.192.38.46
PING 140.192.38.46 (140.192.38.46) 56(84) bytes of data.

--- 140.192.38.46 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 133ms
rtt min/avg/max/mdev = 0.794/1.164/2.771/0.324 ms, ipg/ewma 1.345/1.026 ms
```

Note: RTT=1.164ms

Note that Round Trip Time (RTT) of Task 5 is expected to be significantly higher than that of Task 4.

Task 6: Reconstruct ARP cache by iteratively PING all subnet IP addresses

ARP cache is a table cached by your OS that maps subnet IP addresses to corresponding MAC addresses. Normally, entries of ARP cache are updated upon receiving a reply from a previously sent out ARP request. In implementation, ARP cache will also be updated when receiving PING replies. Let's check it out.

Step1: On VM1, open a new shell script called "arpscan.sh" and type following code in it. You can also edit this file in your host then send it to VM via shared folder (See Task3).

Use your subnet mask instead

```
#!/bin/bash

# ping all ip addresses in the local network
for ip in 10.129.178.{1..254}; do
  # delete old arp records
  sudo arp -d $ip > /dev/null 2>&1
  # get new arp info by ping
  ping -c 5 $ip > /dev/null 2>&1 &
done

# wait for all ping processes to finish
wait

# show scan results (arp table)
arp -n | grep -v incomplete
```

Step2: Change mode of "arpscan.sh" so that you have permission to execute it:

```
[VM1]chmod 774 arpscan.sh
```

Step3: Run the script and view the resulting ARP cache. Can you identify these devices?

Screenshot 5

```
[VM1] sudo ./arpscan.sh
Address                  HWtype  HWaddress           Flags Mask            Iface
10.129.178.232           ether   70:4d:7b:3a:07:6b   C                     eth15
10.129.178.130           ether   38:d5:47:8c:09:6b   C                     eth15
10.129.178.193           ether   e4:a4:71:cc:05:ab   C                     eth15
10.129.178.89            ether   34:c0:59:da:9a:c7   C                     eth15
10.129.178.18            ether   18:65:90:84:6a:d3   C                     eth15
10.129.0.253             ether   00:1a:1e:00:c9:20   C                     eth15
10.129.178.212           ether   2c:1f:23:33:e0:9a   C                     eth15
[VM1]
```

Task 7: Show path to remote machines by TRACEROUTE

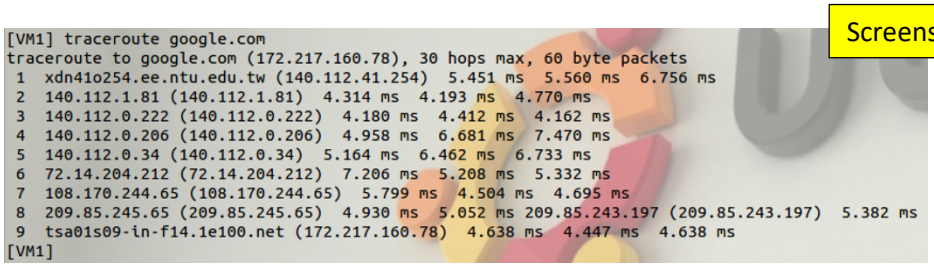
Step 1: On VM1, install "traceroute" by typing:

```
[VM1] sudo apt-get install traceroute
```

Step 2: traceroute to the following destination domain:

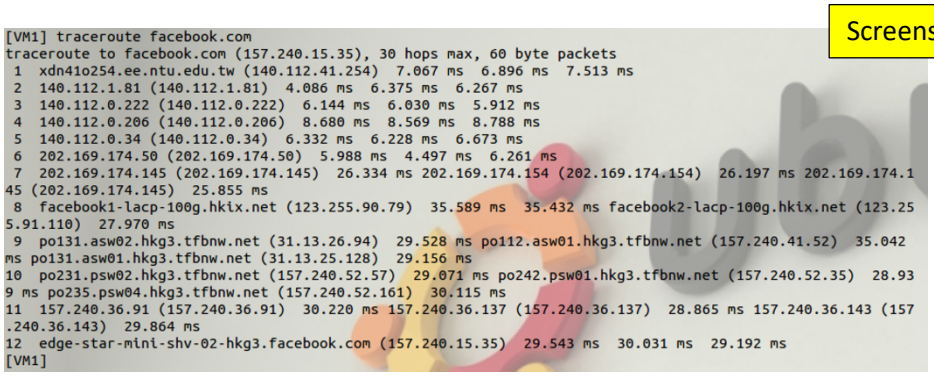
a. google.com

Screenshot 6



b. facebook.com

Screenshot 7



c. cnn.com

Screenshot 8

d. www.ntu.edu.tw

Screenshot 9

Question: In Step2, can you successfully find a path to *cnn.com* or *www.ntu.edu.tw* with traceroute? If not, why? Explain the root cause and your observation in detail. Also provide a method to solve this problem. Besides description, you also need to provide a screenshot that shows your solution. (i.e. a full path from your VM to *cnn.com* or *www.ntu.edu.tw*)

Screenshot 10

Task 8: Testing Communication between VMs on Different Hosts using NAT

Task 8 requires two physical computers as illustrated in Figure 3.

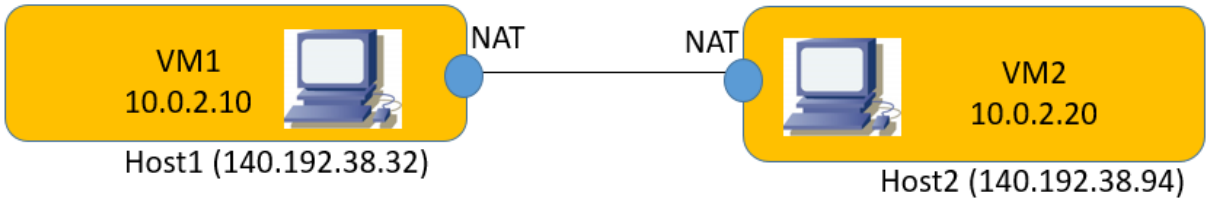
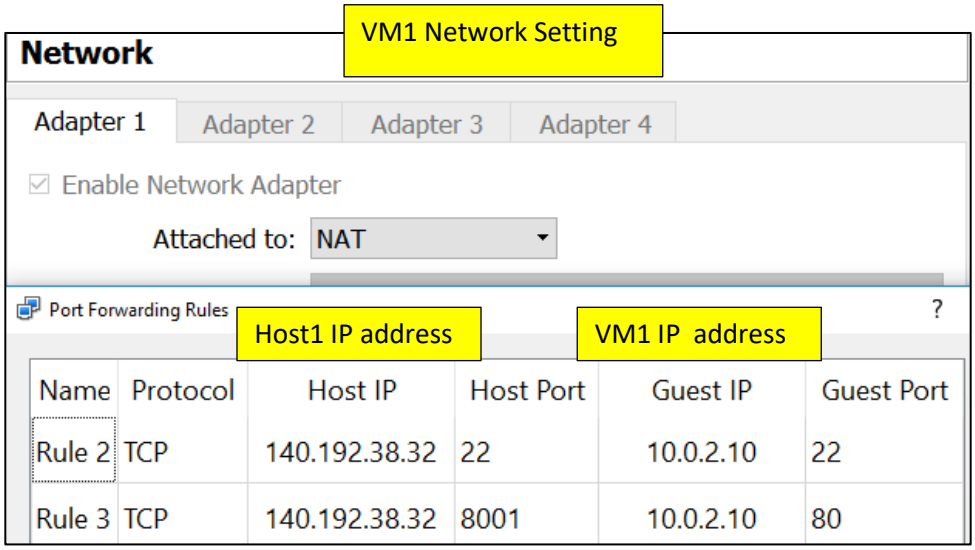


Figure 3. VM Communications between NATed Hosts

Step 1: Provision a VM on each host and choose NAT for the network setting. Also provision port forwarding for the SSH and Web service. The following is the networking setting for VM1. Follow the same procedure for VM2 provision. Note that your Windows firewall may block the incoming HTTP request, and you need to allow the HTTP incoming traffic. Make sure that Host1 and Host2 can ping each other.



Note that port forwarding is not applicable for ICMP so we cannot test ICMP between VMs in this configuration. Also note that we choose 8001 for the web service.

Step 2: Configure an IP address and the default gateway on each VM. Note that the subnet must be 10.0.2.0/24. Also note that your interface may be different from the interface used in the lab00 handout.

```
[VM1] sudo ifconfig eth13 10.0.2.10 netmask 255.255.255.0
[VM1] sudo route add default gw 10.0.2.2
```

```
[VM2] sudo ifconfig eth14 10.0.2.20 netmask 255.255.255.0
[VM2] sudo route add default gw 10.0.2.2
```

Step 3: Edit the default web page to include the VM information and your personal information.

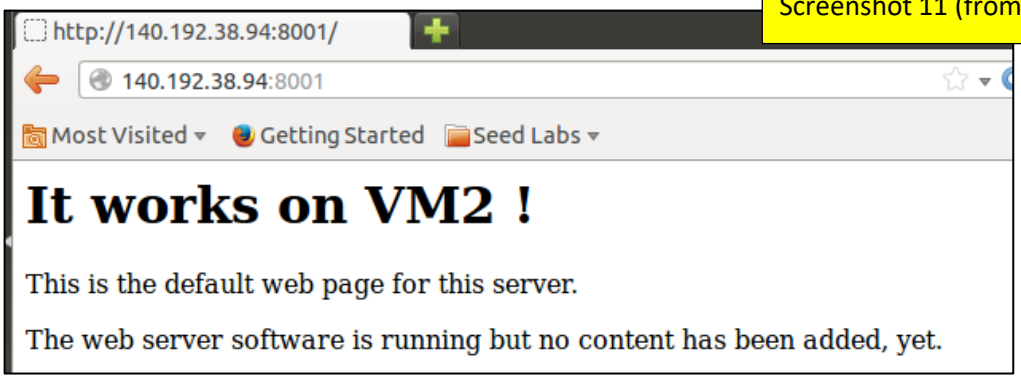
The default web page is located at /var/www/index.html and use an editor to edit this file.

Also add your personal information (name, etc.) o this web page.

```
<html><body>
<h1>It works on VM1!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

Step 4: Start a web browser (FireFox) on VM1 to access a web site on VM2 and vice versa. Note that the URL uses the IP address of the host and the port number is 8001.

Screenshot 11 (from VM1)



Screenshot 12 (from VM2)