Building Your Attack Lab

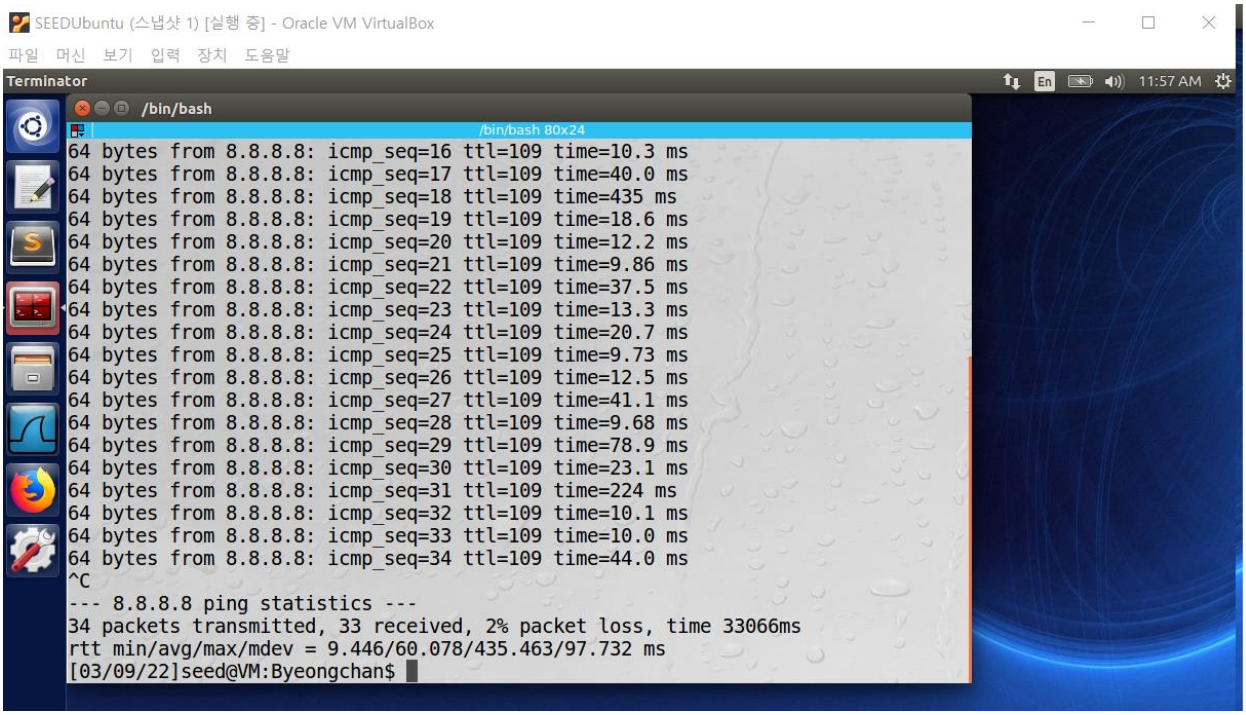
The purpose of this assignment is two-fold. First, we will teach you how to create a simple "Attack Lab" where you can test your exploits on remote machines. Second, you will answer a few questions about the shellshock vulnerability. This assignment is an essential first step for homework 3. Keep in mind that you will need to add and remove machines from this network in the next few weeks, so make sure you understand what you are doing.

You may use this document as a template for your report. Add your additional steps and answer the questions as you work through the lab.

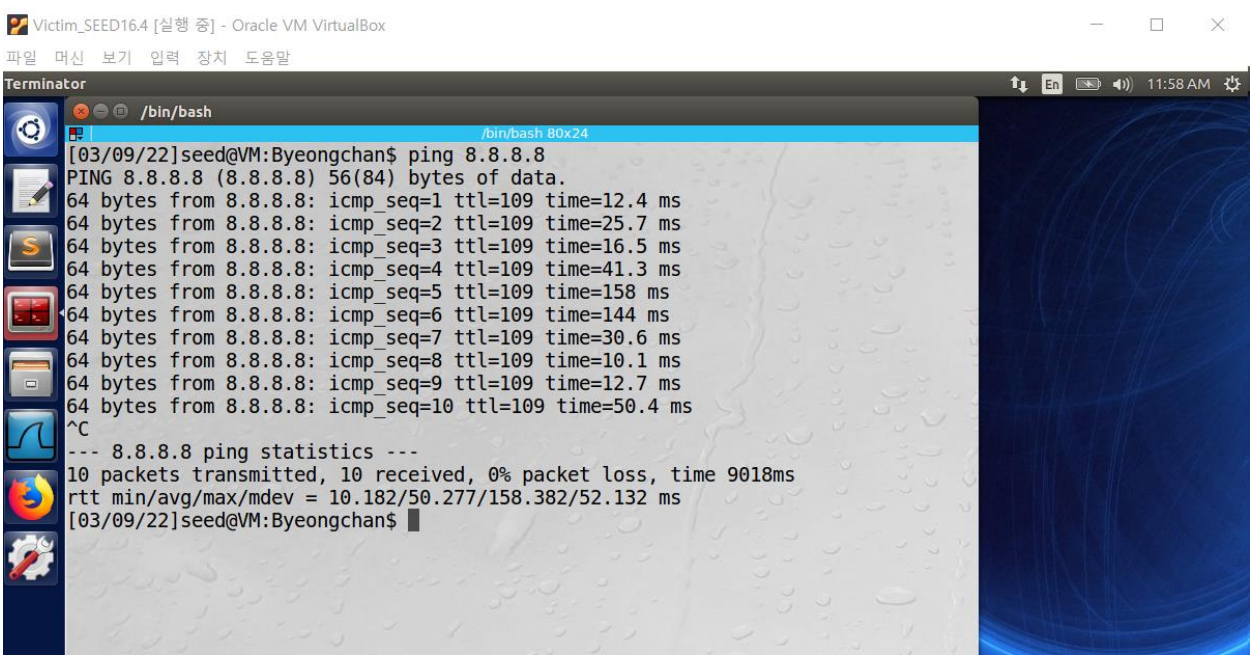
This studio will be graded for correctness.

GATE 1: Clone your VM

1. Go to VirtualBox, locate your SEEDLabs16.4 machine and choose "clone". Name this machine Victim_SEED16.4. (The default options should work, e.g., full clone and current state.)
2. Run the **original 16.4 VM**, start the terminal and type 'ping 8.8.8.8'. This IP address is a primary DNS server for Google DNS, and we use it to verify Internet connectivity. (This will only work if you can reach Google services from the country you are in. If it doesn't work in your country, ping another reachable IP address you know to make sure you have Internet connectivity.) Include a screenshot of the ping output below.

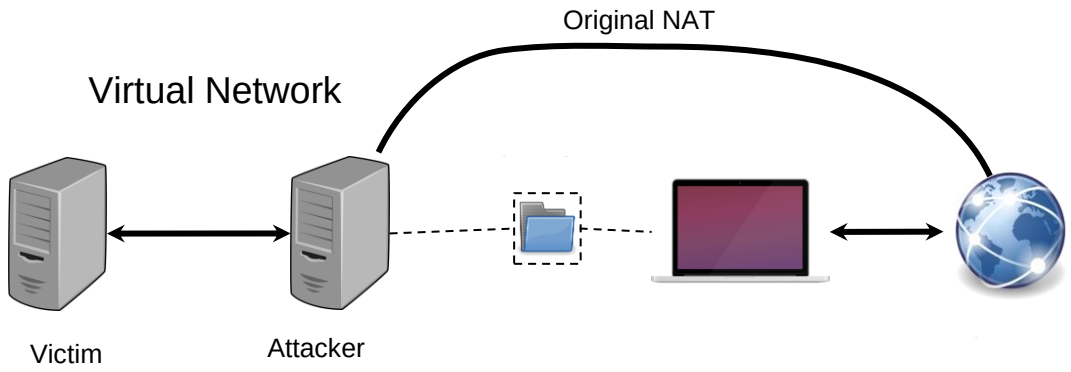


3. Run the **Victim_SEED16.4** VM, and repeat step #2. Include a screenshot of the ping output below.



GATE 2: Creating a local VM network

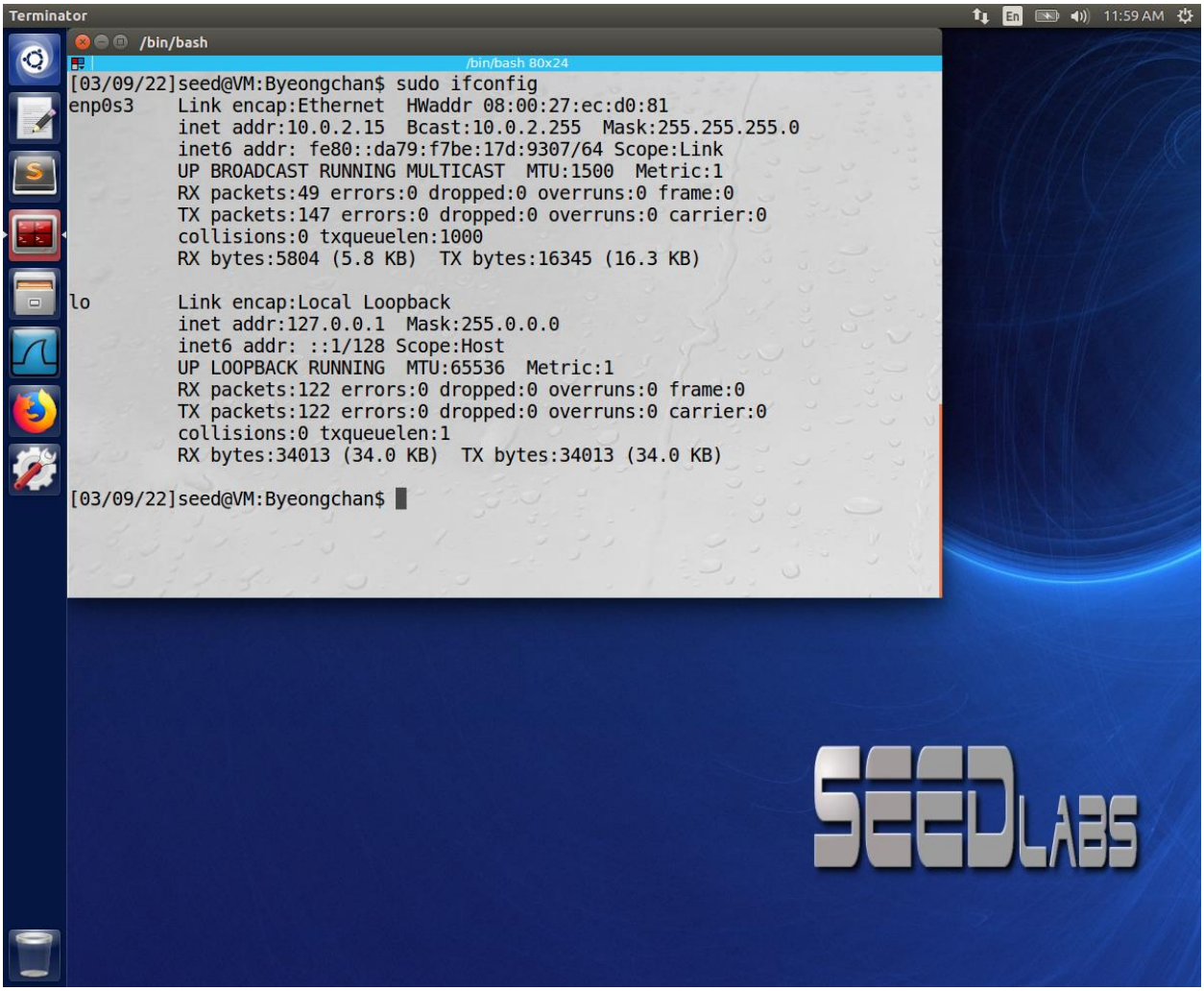
We will now set up an isolated network containing our two VMs as shown in the next figure.



If we do it right, then the cloned VM will not be able to access the Internet when we are done.

<u>Configuration on the original 16.4 machine:</u>

1. Start the machine and type *sudo ifconfig* in the terminal, and include the output below.

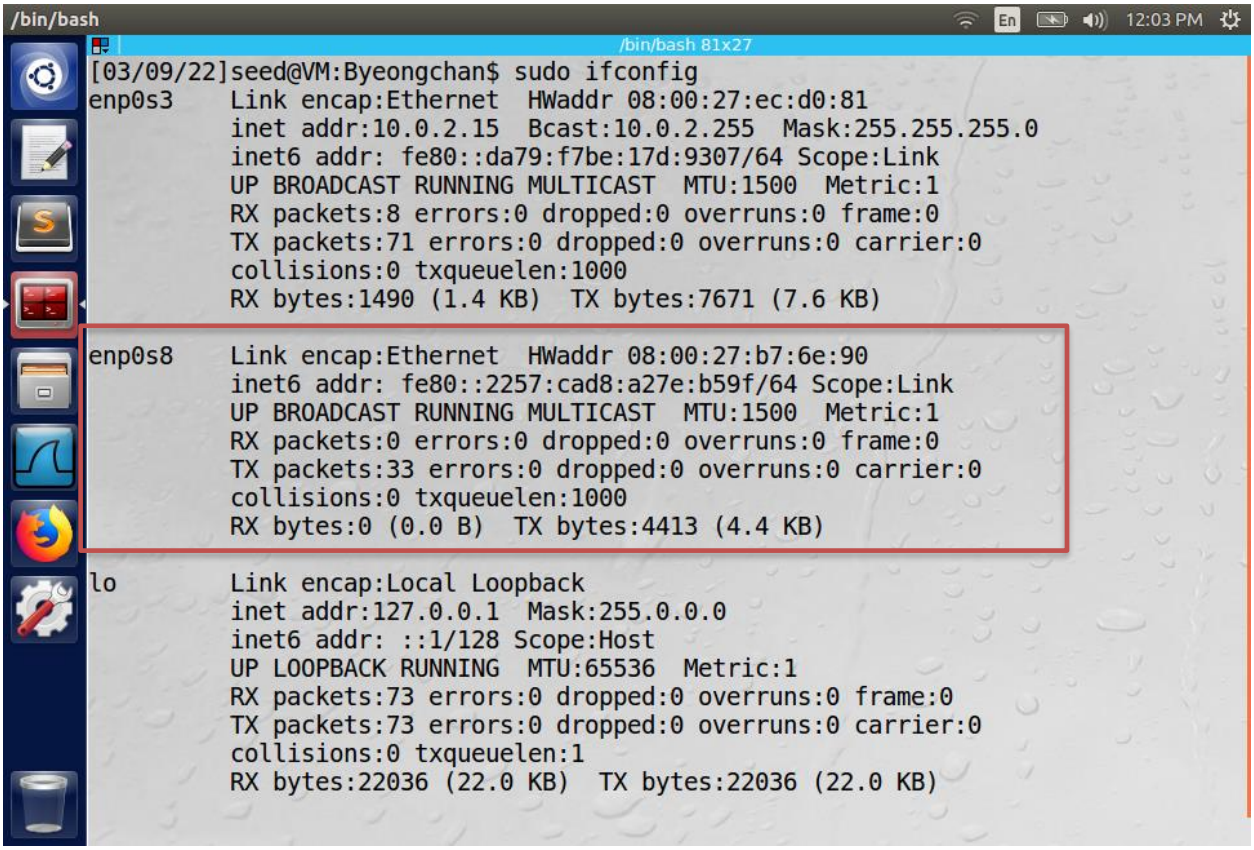


---

2. Shutdown the VM. Open VirtualBox, with this VM selected, and go to Settings->Network.
    a. Go to “Adapter **2**”, and check “Enable Network Adapter”
    b. Change the ‘Attached to’ field to “Internal network”
    c. Enter ‘cse523-internal-network’ as the network name. Click OK.

    These steps will add a second network interface to your original 16.4 VM.

3. Start this machine and run ifconfig again. Paste the output below and mark/bold the new interface (the new interface should start with en***).



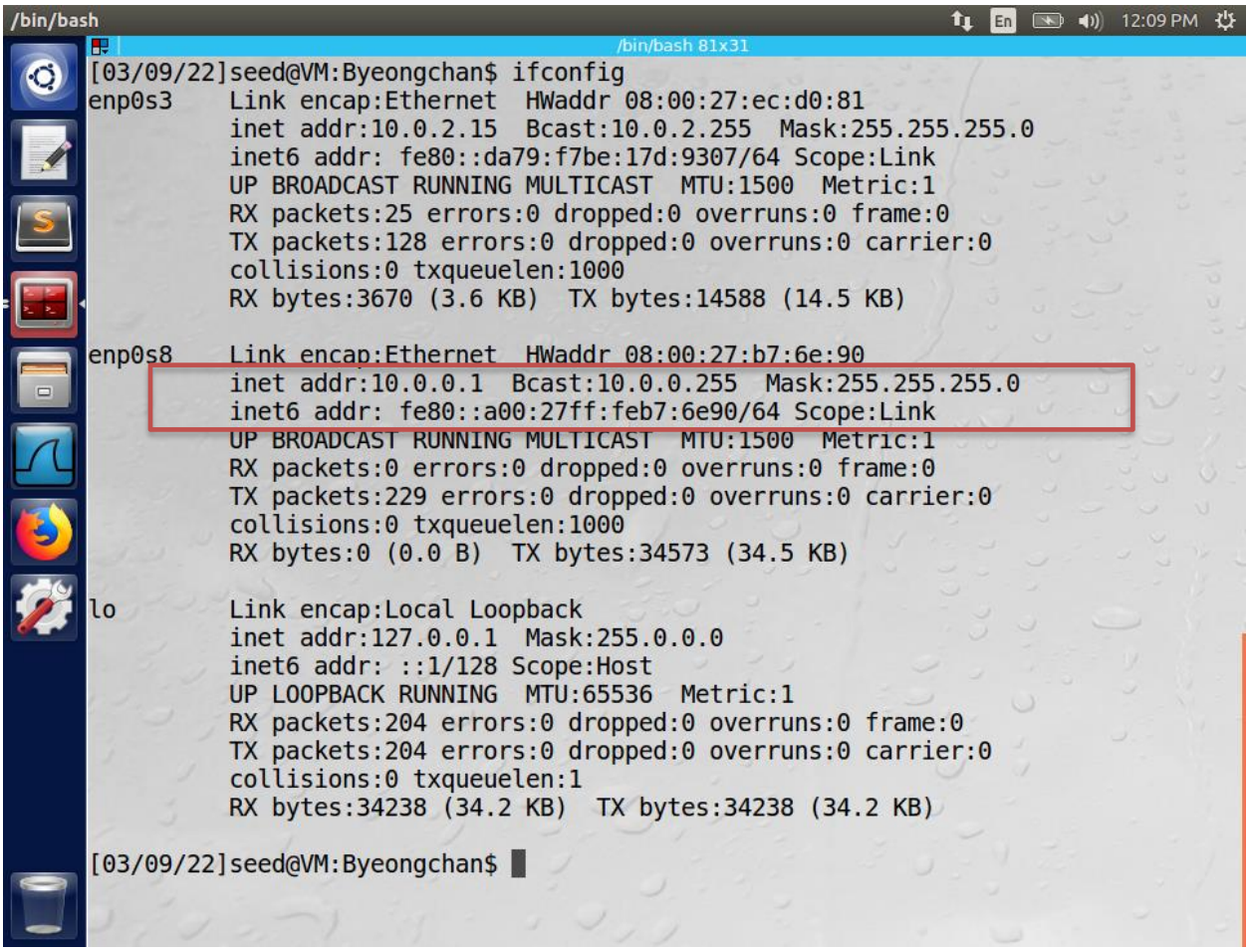
```
/bin/bash                                                    12:03 PM
/bin/bash 81x27
[03/09/22]seed@VM:Byeongchan$ sudo ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ec:d0:81
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::da79:f7be:17d:9307/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1490 (1.4 KB)  TX bytes:7671 (7.6 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:b7:6e:90
          inet6 addr: fe80::2257:cad8:a27e:b59f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4413 (4.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:73 errors:0 dropped:0 overruns:0 frame:0
          TX packets:73 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:22036 (22.0 KB)  TX bytes:22036 (22.0 KB)
```

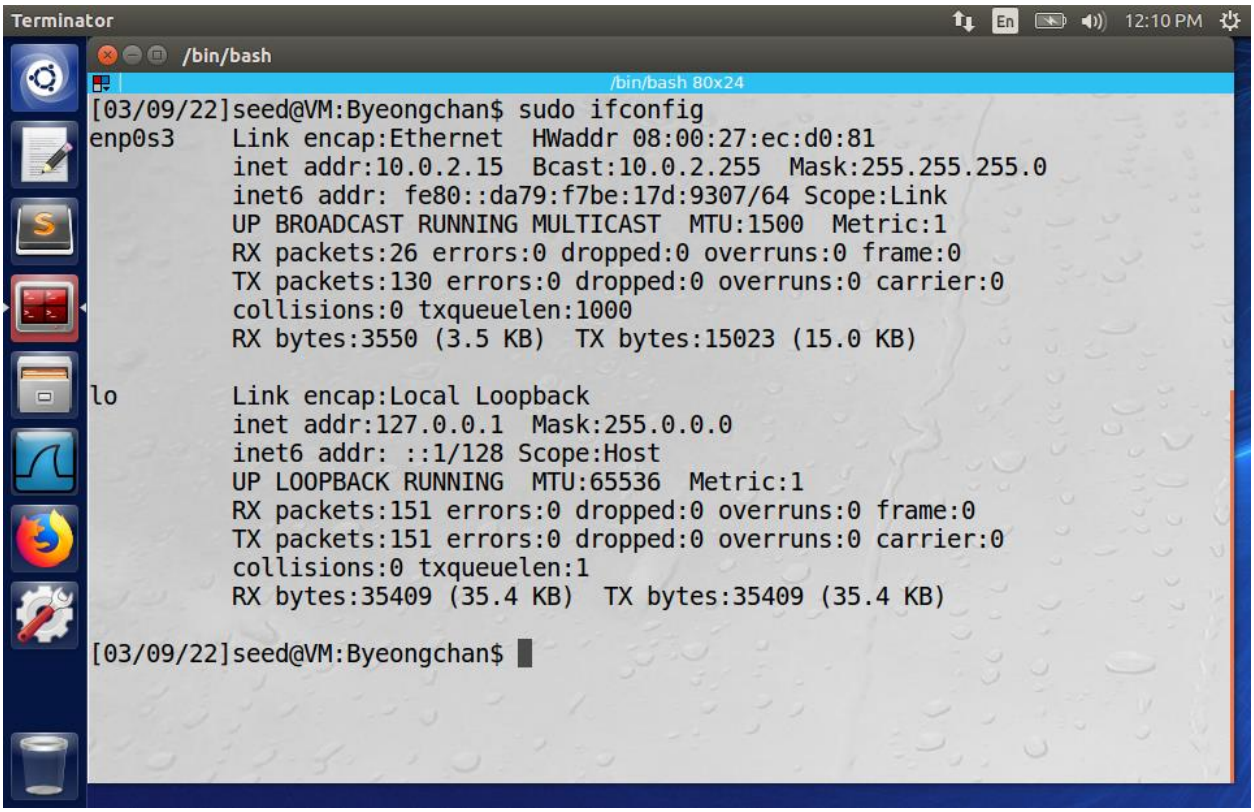
---

4. We will now configure this new interface to use a static IP address: 10.0.0.1 (see **How to configure a static IP address** below). When done, type ifconfig and paste the output below. Mark/bold the changes.



Configuration on the cloned 16.4 machine (Victim_SEED16.4):

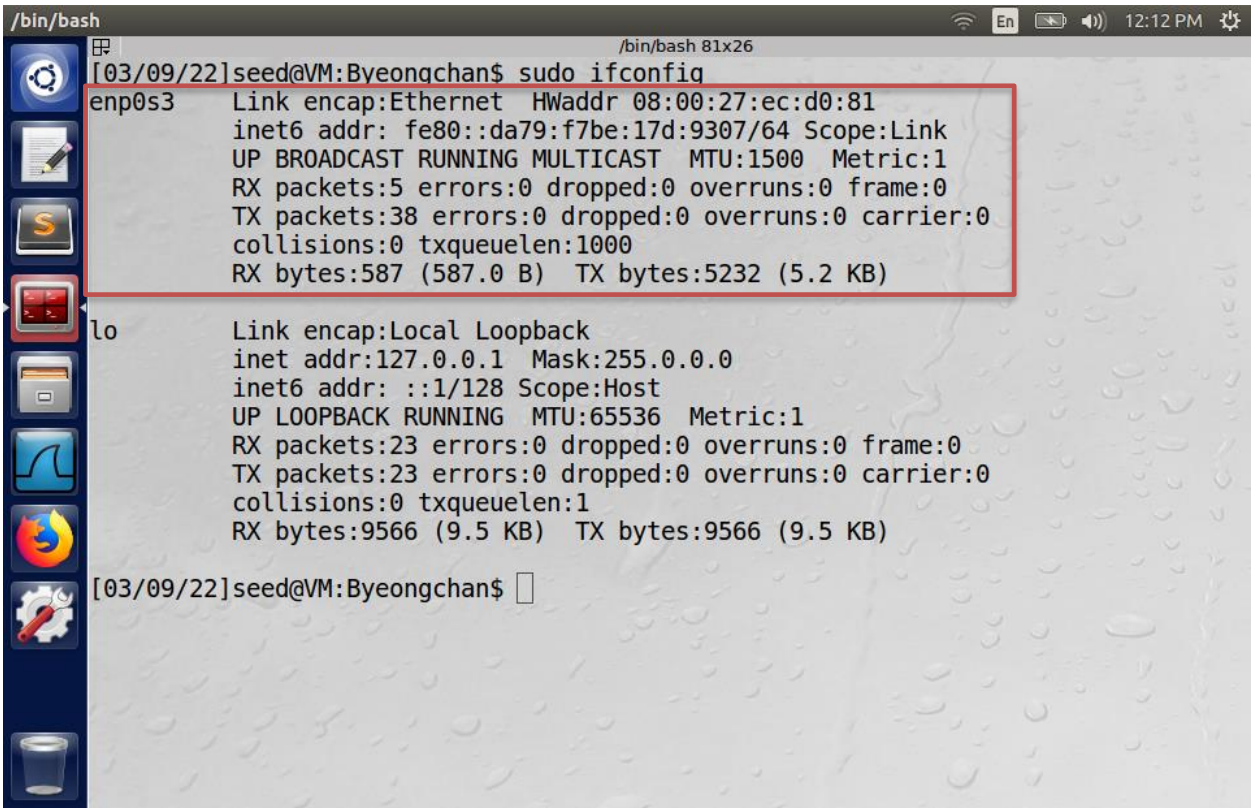
5. Start the machine, type *sudo ifconfig* in the terminal, and include the output below.



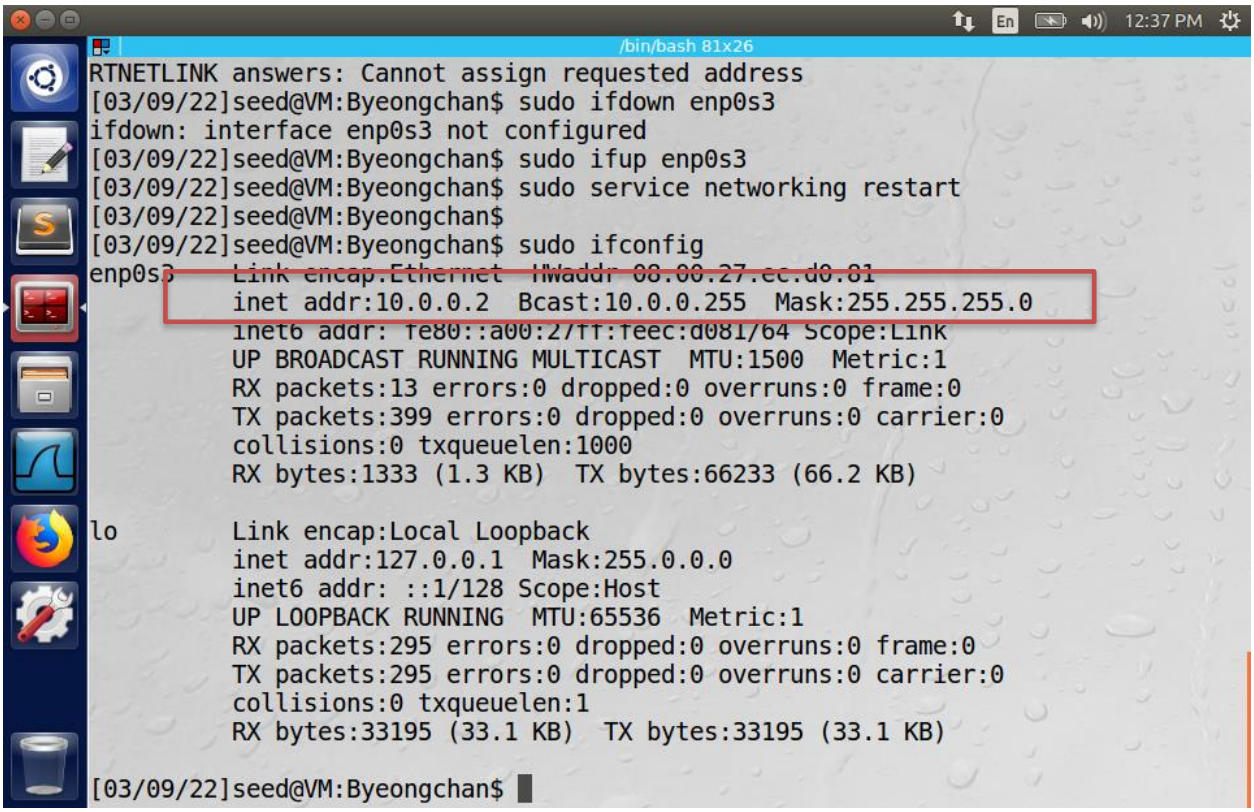
---

6. Repeat step #2 (of part 2), but configure the Internal network on Adapter 1. (There is no need to enable adapter 2).

7. Start this machine and run ifconfig again. Paste the output below and mark/bold the interface corresponding to Adapter 1 (the interface should start with en***).



8. configure this interface to use a static IP address: 10.0.0.2. When done, use ifconfig and paste the output below. Mark/bold the changes.



<u>How to configure a static IP address</u>

- You will need to modify /etc/network/interfaces (e.g., sudo nano …) to configure your network interfaces. Your configuration should include a new entry like the following (don't forget to change the [interface name] and [new static IP])

  auto [interface_name]
  iface [interface_name] inet static
  address [new static IP]
  netmask 255.255.255.0

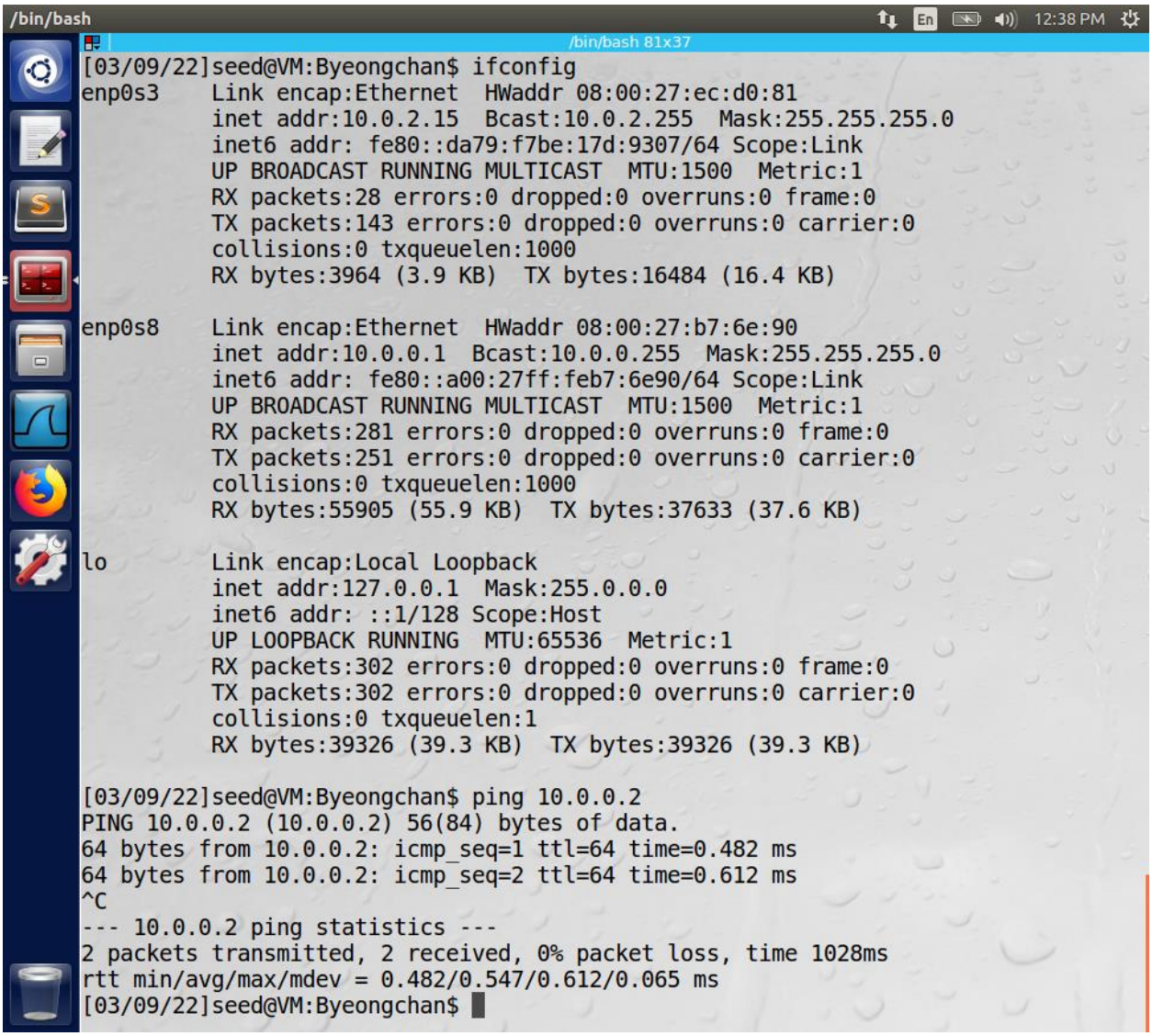
  - you may have to run the following commands in order to reset network interfaces
    - *sudo ifup [interface_name]*
    - *sudo service networking restart*

GATE 3: Check your setup

Now, ping 8.8.8.8 from each machine to make sure you get the expected behavior. Add the two screenshots below. State which screenshot was taken from which machine.

Conclude this part of your report with screenshots of your VMs' IP addresses (using ifconfig), followed by a ping to the other machine. Take screenshots of the two machines pinging each other. Indicate which screenshot was taken from which machine.

Below is the original machine



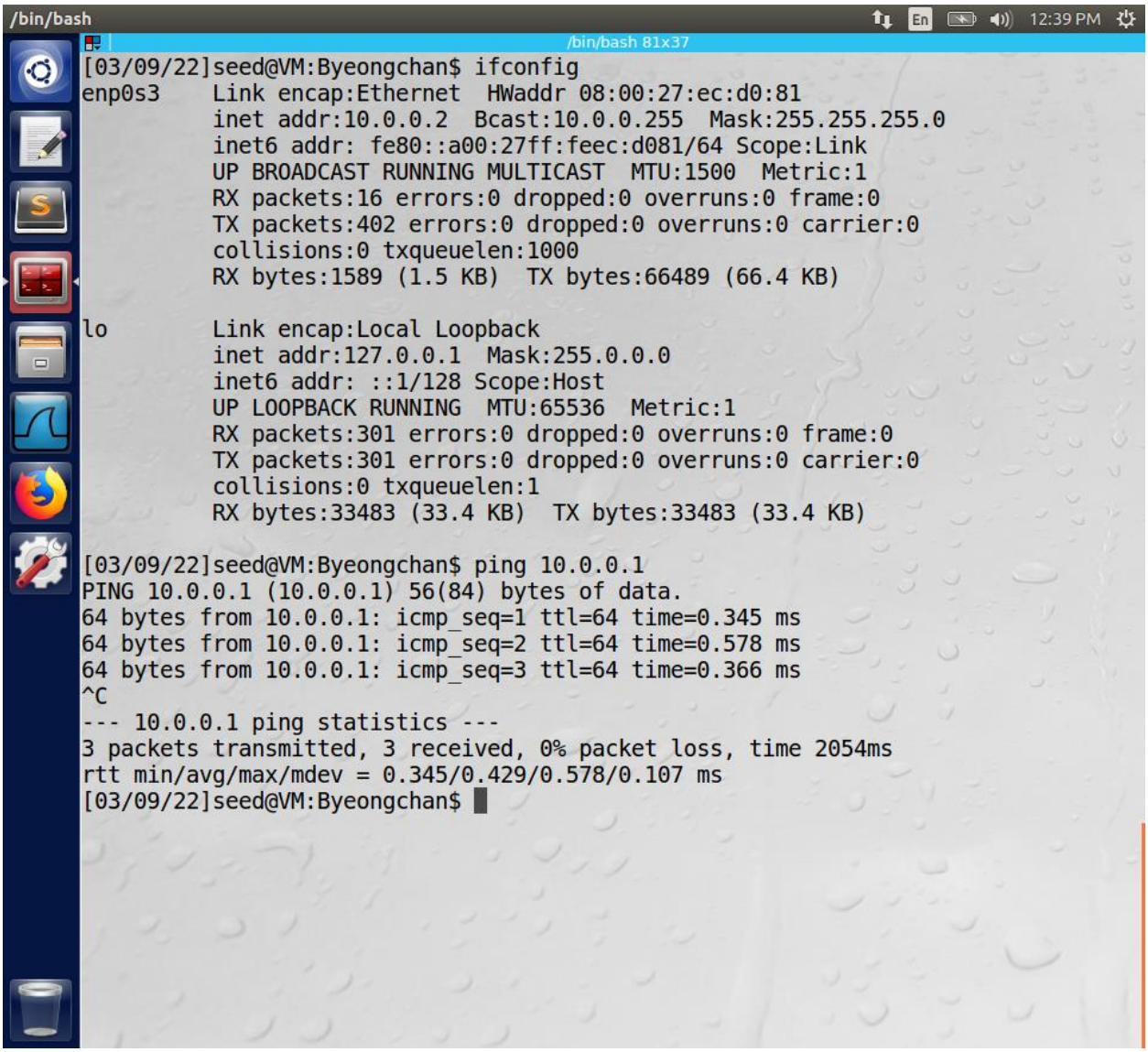
```
[03/09/22]seed@VM:Byeongchan$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ec:d0:81
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::da79:f7be:17d:9307/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28 errors:0 dropped:0 overruns:0 frame:0
          TX packets:143 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3964 (3.9 KB)  TX bytes:16484 (16.4 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:b7:6e:90
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb7:6e90/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:281 errors:0 dropped:0 overruns:0 frame:0
          TX packets:251 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:55905 (55.9 KB)  TX bytes:37633 (37.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:302 errors:0 dropped:0 overruns:0 frame:0
          TX packets:302 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:39326 (39.3 KB)  TX bytes:39326 (39.3 KB)

[03/09/22]seed@VM:Byeongchan$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.482 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.612 ms
^C
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1028ms
rtt min/avg/max/mdev = 0.482/0.547/0.612/0.065 ms
[03/09/22]seed@VM:Byeongchan$
```

Below is the victim machine



You will need to repeat many of these steps in the next few weeks when adding new machines or creating different networks. A detailed and correct report would be a helpful resource!

GATE 4: Prepare for HW3

What is the shellshock bug? Briefly explain the root cause of the vulnerability and what made this attack so dangerous.

In a Linux system, we can set variables that we want to use throughout the system. It is called the 'Environment variable'. We can make normal variables and also, we can make variables with the function definition. There's a problem when setting a variable with a function definition. When you set a function into a variable, commands after the function definition are also executed. It's not intended. Therefore, if you add commands at the end of the environment variable with a function definition, you can execute the commands whatever you want.

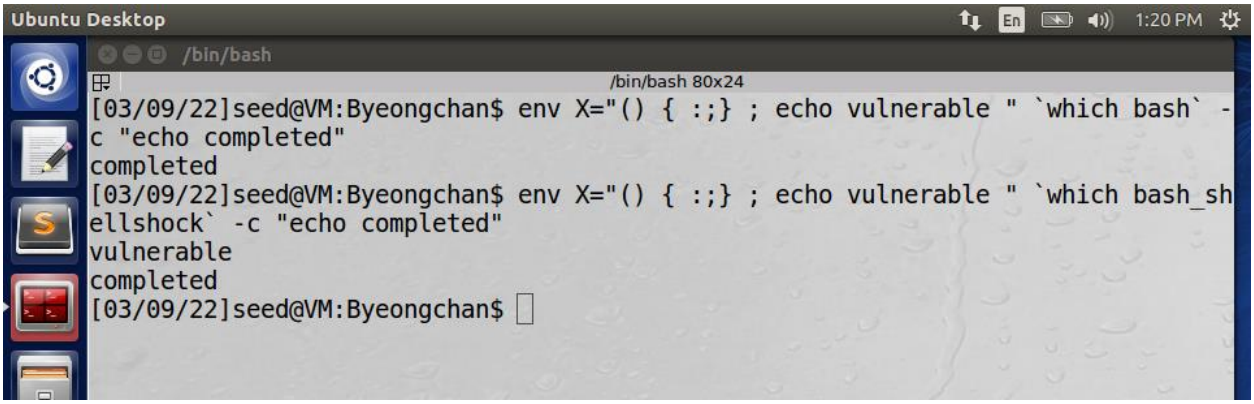
You can use the following script to check if your system is vulnerable:

```
env X="() { :;} ; echo vulnerable " `which bash` -c "echo completed"
```

Based on what you learned in the lecture, briefly explain why this script is helpful.

There is a function definition which is followed by the 'echo vulnerable' command. If your bash is vulnerable and then the following command will be executed. If your bash is secure and then the following command will not be executed.

Start your **Victim_SEED16.4** VM, and run this script twice. Once to check if `bash` is vulnerable and a second time to check if `bash_shellshock` is vulnerable (by replacing bash with bash_shellshock in the script.) Take a screenshot showing the output of the two tests and paste it between the lines.



If successful, you should see two different outputs. Briefly explain the why.

When executing with normal ‘bash’, it can prevent to execute after the function definition. However, when executing with abnormal ‘bash’, if cannot prevent to execute the following command after the function definition.