Exercise 1

Networks and Systems Security II (CSE354/554)

Winter 2025

Total points: 55

Deadline: Feb 11, 2024 (2359hrs)

There are two tasks in this exercise. You need to attempt both and submit a report carrying screenshots, commands you ran and their descriptions

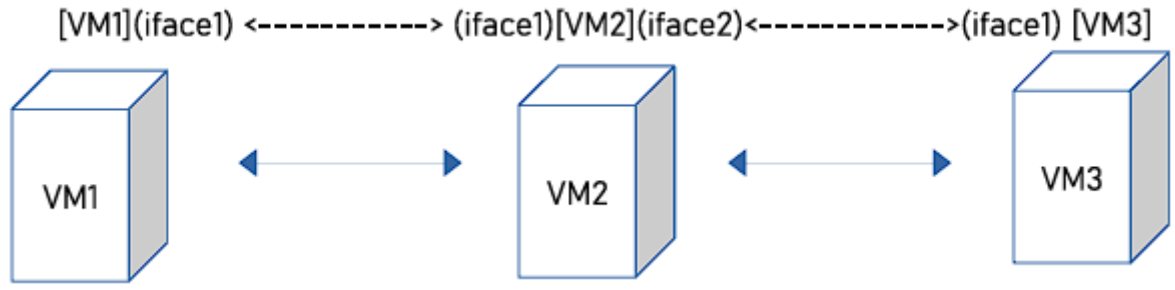
Task 1 (total points: 30)

1. One VM running FreeBSD 14.0-RELEASE with no X window. You could use a very lean configuration -- e.g. 1 core and 1 GB RAM. This should be acting as a L3 forwarding firewall. You need to enable the IP routing for the same. For this exercise we would be relying on FreeBSD pf firewall.

2. Two other VMs, running any OS of your choice (Linux or FreeBSD) with as minial configuration as possible.

3. The three VMs must be connnected to one another using virtual networks, separate from the ones used to connect each of the VMs to the Internet.

Objective: The exercise is designed to make you play with FreeBSD pf (an industry standard firewall framework) to do traffic filtering (a form of network DAC). You would require configuring the three VMs using the following configurational setup



a. Assign private IP addresses to each interface, such that the VMs can ping one another. VM1 (iface1) and VM2 (iface1) should be in the same subnet. VM2(iface2) and VM3 (iface1) should be in the same subnet.

b. Setup IP forwarding on VM2 such that VM1 can ping both interfaces IPs of VM2 and VM3. Make sure traffic from VM1 to VM3 goes via VM2 (can be seen via traceroute).

c. On VM3 add another interface with which it can access the Internet. Install a HTTP server (e.g. Apache, lighttp or nginix) on the VM3. Create a web directory and add some files to it.

d. You need to use pf on VM2 and configure bi-directional Network Address Translation on VM2. Thus, when VM1 access the webserver on VM3, or when VM2 access the webserver on VM3, in both cases you should observe connections coming from VM2. Similarly when responses go back to VM1, it should appear that they are arriving from VM2 and not VM3. You need to show screenshot outputs taken from Wireshark, showing the bidirectional NATing.

e. Enable webserver process on port 80 and 443. Configure the firewall on VM2 to restrict access only to those ports and not allow anything else. To test correct functionality you could additionally install SSH server on VM3 and try to access it. Correct firewall functionality should restrict access to port 22 (SSH).

f. Enable webserver process on port 80 and 443. Configure the firewall on VM2 to restrict access only to those ports and not allow anything else. To test correct functionality you could additionally install SSH server on VM3 and try to access it. Correct firewall functionality should restrict access to port 22 (SSH).

The following links are a good starting point to learn more about FreeBSD pf:
https://docs.freebsd.org/en/books/handbook/firewalls/
https://www.digitalocean.com/community/tutorials/how-to-configure-packet-filter-pf-on-freebsd-12-1

The book of pf: Chapter 2,3 and 5. (see reference/reading material)

Grading rubric:

1. Writeup describing the following:

a. The commands used to enable routing (IPv4 forwarding) on VM2 with screenshot showing that it was able to successfully do so. You also show a screenshot where you run traceroute from VM1 targeted to VM3 - (10 points).

b. The commands you issued on VM2 to enable bi-directional NAT and the subsequently modified NAT tables (shown via appropriate screenshots) (5 points).

c. The wireshark/pcap/tcpdump output snippet (not the entire capture) on all the three machines that shows the bi-directional NAT in action. On VM3 it should appear that requests originate from VM2, while on VM1 he target of the request to port 80/443 is towards VM2 (which would internally perform the NAT and send the packets to VM3) (15 points)

Task 2 (total points: 25)

This task requires setting appropriate file permissions on the VM3.

a. On VM3 now you need to create one more user - temphttp. The webdirectory on the server should be onwed by temphttp with no permissions for the webserver process to read from it.

b. Can you utilize the `setuid' bit to set the permissions of the webserver process to be able to write to the webserver directory? If so,then please enumerate the steps with the screenshots for the outputs.

c. Can you use Linux ACLs to grant specific access read access to the webserver process? If so, then please enumerate the steps and show the output with the screenshots for the outputs.

Useful links:

https://linuxconfig.org/how-to-manage-acls-on-linux

https://www.tecmint.com/secure-files-using-acls-in-linux/

https://bencane.com/2012/05/27/acl-using-access-control-lists-on-linux/

Grading rubric:

1. Write-up describing the following:

a. Why you can or can't access the web server directory even with`setuid' bit set **(10 points)**.

b. Screenshots of commands you executed and the output **(10 points)**.

c. Screenshots of commands and outcomes in case you are able to access the webserver directory after appropriate `setuid' bit values **(5 points)**.