Lab 07: IP addressing and Static routing

OSL, Mon Oct 8, 2012

Objective:

1. Learn how *IP address configuration* and *routing* are setup in Linux.

General instructions:

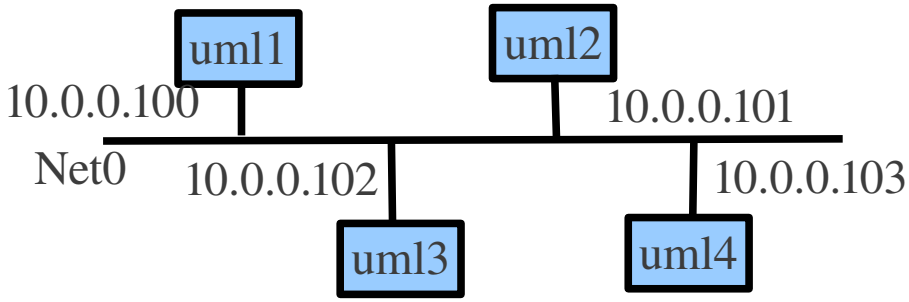
1. This lab is to be done **individually.**
2. Create a directory called <rollnumber>_lab07. As you proceed with the lab instructions below, note down observations or relevant output from whatever you do in a file named ”lab07.txt” using a text editor.
3. Also add to this directory any trace files collected. You will find more details of this in the specific exercises.

Lab Instructions:

This lab is fully based on VNUML. We will use tcpdump extensively in most of the below exercises. A few points to note.

1. To run tcpdump in the background to free up the console for other commands, use “tcpdump -enx -w trace.out &”.
2. If you wish to terminate tcpdump when running it in background, you will need to kill the tcpdump process. Use “ps ax | grep tcpdump” to get the process number and kill” command to kill it.
3. You can view the output of tcpdump by executing the following command “tcpdump -r trace.out | less”. If you wish to view addition information like the link layer headers, you can use -e option.

Exercise 1: IP address mis-configuration



Create a simple topology as shown in the above figure using vnuml (name the file lab07-addr.xml). We have 4 machines connected to a LAN segment. Each interface of a machine is assigned an IP address as shown. Bring up the above virtual network using the command: “ vnumlparser.pl -t lab07-addr.xml -v”. *In case you want to release the simulator for whatever reason, you can do so using the command “vnumlparser.pl -d lab07-addr.xml -v”.*

Use the “ifconfig” command to change the IP adress of uml2 from 10.0.0.101 to 10.0.0.100. Essentially, we are assigning duplicate addresses to two machines uml1 and uml2. Ensure that the

subnet mask of all the machines is 255.255.255.0. If the subnet mask is different, use the ifconfig command to make it 255.255.255.0.

Also, we will like to start the below experiments with a clean slate. Ensure that at all machines, there are no arp entries for any other hosts. Delete the entries if necessary using arp -d.

Run tcpdump on all the hosts in the background and then do the following experiment.

Part-1: Pinging a host from machines with duplicate IP addresses

> Ping uml3 from uml1 (use the -c option to limit the number of packets to 10). Now ping uml3 from uml2 (again use -c option to limit number of packets to 10). Start the second ping from uml2, only after some 4-5 ping packets got sent from uml1. Once ping finishes, check the arp table at the different hosts and go through the tcpdump outputs on all the hosts. How many ping replies did uml1 and uml2 recieve? Explain all your observations in the lab report. (See how a connection can be hijacked?)

Again clear out all arp entries and run tcpdump in background and do the following experiment.

Part-2: Pinging the duplicate IP address from different hosts

> Ping (-c set to 3) 10.0.0.100 from 10.0.0.102. After this ends, ping 10.0.0.100 from 10.0.0.103. Again go through the tcpdump results to figure out what happened and explain it crisply in the report. In the first ping, which host replied to 10.0.0.102? and in the second ping, which host replied to 10.0.0.103. If uml2 is a malicious host, what does it need to do to capture the ping requests so that it can reply to them?

Exercise 2: Sub-net mask mis-configuration

We will continue with the same vnuml configuration as earlier. in this exercise, change the IP adress and subnet mask of the various machines as shown.

uml1: 10.0.0.100 255.255.255.240

uml2: 10.0.0.101 255.255.255.0

uml3: 10.0.0.102 255.255.255.0

uml4: 10.0.0.120 255.255.255.240

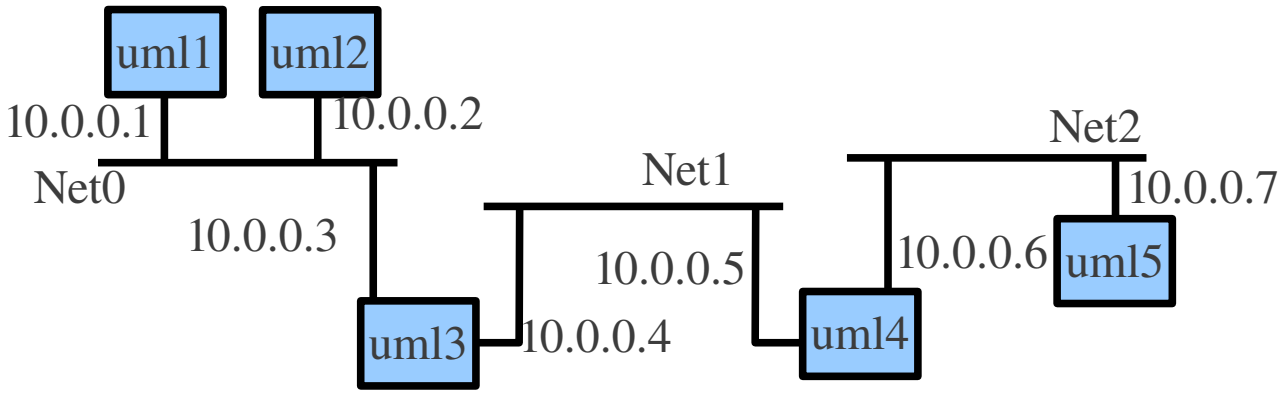
We have purposefully assigned an incorrect subnet mask to machines uml1 and uml4. It helps to look at the binary representation of the masks and IP addresses. Use tcpdump to capture packets as appropriate and do the following experiments.

> **Expt1:** From uml1 (which has incorrect address mask) ping uml2 (which has correct address mask). From uml4 (which has incorrect address mask) ping uml2 (which has correct address mask). Specifically why was one machine with incorrect mask able to ping while the other which also has an incorrect mask was not able to. You may want to run tcpdump on uml2.
>
> **Expt2:** From uml2 which has correct address mask, ping uml1 and then uml4, both of which have incorrect mask. What was the exact reason for ping failure in this case? Compare it with the previous case of expt 1. You may want to run tcpdump on uml1 and uml4.

After you are done playing with the simulator, do not forget to release it using the "-d" option.

Exercise 3: Manual Routing

In this exercise we will use a different vnuml configuration (lab07-routing.xml). Create the topology shown below in vnuml.



This topology is very similar to what you did as part of bridging. Some of you may have encountered this problem where you could ping one host from another but not the other way. We will debug this problem and rectify it in this exercise.

1. From uml3, ping 10.0.0.5. You will notice that it will not work. Use tcpdump to debug why this ping is not working and report the answer in the report.
2. Now use the "-I" option of ping to make this ping work. Usetcpdump to debug. Again report as to why ping works in this case.
3. Now we will try ping in the reverse direction. Ping 10.0.0.4 from uml4. With or without the "-I" option, ping will not work. Why?
4. Go through the man pages of the "route" command and fix the problem using the "route" command. After you are done, ping should work either way with and without the "-I" option. Note down in the report how you fixed the problem (exact command used).

If anything goes wrong, you can just reboot the virtual machines; or better still, try to restore order by doing "ifconfig eth1 down" followed by "ifconfig eth1 10.x.y.z netmask 255.255.255.0 up". You can also learn how to delete routing entries, by using "route del" with the appropriate arguments.

Submission instructions

The directory named <rollnumber>_lab07 that you will submit should contain the following files:

1. lab07.txt
2. lab07-addr.xml
3. lab07-routing.xml

Now tar it as follows:
tar -zcvf <rollnumber>_lab07.tgz <rollnumber>_lab07/

Submit the file <rollnumber>_lab07.tgz via moodle for grading.