CSP334: Computer Networks
Lab Assignment No 2
Assignment on Linux Networking Commands

Date of submission : 11:59 hrs, 23rd August 2018

Date uploaded :17th August 2018

Instructions:

- **For each of your answer paste a screen dump of your screen in support of justification of your answer, wherever applicable**

1. Examine the following files in Linux and find out what is the purpose served by each. Write at least two sentences mentioning the purpose of each.
   - /etc/hosts
   - /etc/sysconfig/network
   - /etc/sysconfig/network-scripts/ifcfg-eth0
   - /etc/default-route
   - /etc/resolv.conf
   - /etc/nsswitch.conf
2. Display the file /etc/services on your screen, using appropriate Linux command. What is the use of /etc/services file? Which layer in the TCP/IP protocol stack do you think would make use of this file ? Are the port numbers shown in this file *well-known* port numbers or *ephemeral* port numbers ? Why are they so ? Give appropriate reasoning for your answer.
3. Read the man pages for the following programs:
   - arp
   - arping
   - ifconfig
   - tcpdump
   - ping
   - netstat
   - route

   Find out the the purpose of each of these commands. In those cases whereever applicable, list out the application layer, transport layer and the network layer protocols used by each command. Prepare a table with the following columns to answer your question viz. command_name, Purpose, Transport_layer_protocol_used, Network_layer_protocol_used.
4. This exercise is a simple exercise that only requires you to capture the *tcpdump* traffic. The problem requires you to either use two virtual machines on your laptop or two different machines in the computer lab - ask the administrator for the host name of both the machines, if so. Then run the *tcpdump* command on one machine say PC1 (saving the output for your lab report) so that it monitors all the packets that contain the IP address of PC2 only and none else. Next, open a new terminal window on PC1 and execute a *ping* command to PC2. It may be necessary to press Ctrl-C to terminate the *tcpdump* session. It may sometimes be best to simply redirect the output of *tcpdump* straight to a file and view it afterward with the *more* command or a text editor. Find out how can you do so.

5. Run the command tcpdump −enx −w exe5.out. Do you see any output on the screen ? Why ?

6. This question is in continuation of the question no 5. Run telnet remote_host. remote_host is the host name of either another virtual machine in your machine or it is the host name of any other machine in the network used in the lab (Ask the lab technical suport staff about the name of other machine). This command would generate some TCP traffic. After you login the remote machine, terminate the *telnet* session and terminate the *tcpdump* program.

   Next, you will use *wireshark* to open the packet trace captured by *tcpdump* and analyze the captured packets. To do this, run *wireshark -r exe5.out &*. The *wireshark* Graphical User Interface (GUI) will pop up and the packets captured by *tcpdump* will be displayed. For your report, you need to save any one of the packets that contain the *link, IP, and TCP* headers. Carry out the following instructions.

   - Click on a TCP packet from the list of captured packets in the wireshark window. Then go to the Edit menu and choose Mark Frame.
   - Go to the File menu and choose Print. In the *Wireshark:Print* dialog that pops up, check *File, Plain Text, Expand all levels, Print detail* and supress *unmarked frames*. Then, enter the output text file name, e.g., *headers.txt*, and click the OK button. The marked packet is now dumped into the text file, with a detailed list of the name and value of every field in all the three headers.

   Now answer the following questions:

   a. Draw the format of the packet you saved, including the *link, IP, and TCP headers* (See Figs in the handouts given to you for reference), and identify the value of each field in these headers. Express the values in the decimal format.

   b. What is the value of the protocol field in the IP header of the packet you saved? What is the use of the protocol field?

7. In a manner similar to the previous exercise, now run *tcpdump* to capture an ARP request and an ARP reply and then use *wireshark* to analyze the frames. If there are no *arp* requests and replies in the network, generate some using *arping* a remote machine. After you see several ARP replies in the *arping* output, terminate the *arping* and the *tcpdump* program. Open the *tcpdump* trace using *Wireshark -r exe7.out &*. Print one ARP request and one ARP reply using *wireshark*. Now answer the following questions:

   a. What is the value of the frame type field in an Ethernet frame carrying an ARP request and in an Ethernet frame carrying an ARP reply, respectively?

   b. What is the value of the frame type field in an Ethernet frame carrying an IP datagram captured in the previous exercise?

   c. What is the use of the frame type field?

8. Explain briefly the purposes of the following *tcpdump* expressions.

   a. tcpdump udp port 520

   b. tcpdump -x -s 120 ip proto 89

   c. tcpdump -x -s 70 host ip addr1 and (ip addr2 or ip addr3)

   d. tcpdump -x -s 70 host ip addr1 and not ip addr2

9. Start *tcpdump* in a command window to capture packets between your machine and a remote host using: *tcpdump -n -nn host your_host and remote_host*. Execute a TCP utility, *telnet* for example - as in the problem before, in another command window. When you see a TCP packet in the *tcpdump* output, terminate *tcpdump* and save its output. Now answer the following question:

   a. What are the port numbers used by the remote and the local computer?

   b. Which machine′s port number matches the port number listed for *telnet* in the */etc/services* file?

   Note: In case *telnet* is not listed in the */etc/services* file, use *ssh*.

10. Start *tcpdump* in one command window using *tcpdump -n -nn host your_host and remote_host*. Then, *telnet* to the remote host from a second command window by typing $telnetremote_host$. Again issue the same $telnetremote_host$ command from a third command window. Now you are opening two telnet sessions to the same remote host simultaneously, from two different command windows. Check the port numbers being used on both sides of the two connections from the output in the *tcpdump* window. Save a TCP packet from each of the connections. Now answer the following questions:

    a. When you have two telnet sessions with your machine, what port number is used on the remote machine? Are both sessions connected to the same port number on the remote machine?

    b. What port numbers are used in your machine for the first and second telnet, respectively?

    c. What is the range of Internet-wide well-known port numbers? What is the range of well-known port numbers for Unix/Linux specific service? What is the range for a client port number? Compare your answer to the well-known port numbers defined in the /etc/services file. Are they consistent? In case they are not, try to discuss amongst peers and specify your view of the reason why they are not.

    Note: In case *telnet* is not listed in the */etc/services* file, use *ssh*.