CS393 / CS682 Lab 0
TCP/IP and Linux Socket Programming

Introduction:
TCP/IP protocol suite is the core of Internet and it is necessary to understand how these protocols work together, their strength and weaknesses, and how they are used to secure networks. TCP/IP comprises a set of powerful protocols, but they have many drawbacks when it comes to security. A thorough understanding of TCP/IP suite is vital to proper understanding of network security issues.

This lab is divided into four parts. Starting from basics to advanced topics in TCP/IP. This lab cannot be dropped and must be done individually.

Requirements:
- Read about TCP/IP protocol from
  o Books 24x7
- Read about Linux, a number of books are available in books 24x7 about Linux.
- Read about Linux socket programming from:
  o Linux Socket Programming By Example by Warren W.Gay. Again this book is also available in Books 24x7
  o http://www.uwo.ca/its/doc/courses/notes/socket/

Your Task:
Part 1: Basics of TCP/IP
Answer the following question:

1) Name all the protocols that are a part of TCP/IP suite, their purpose, at what OSI layer they function and the protocols that each protocol is dependent on.

2) Explain the following terms (No more than 10 lines for each term), and describe their functions:
   a. Subneting
   b. Subnet mask
   c. IP address
   d. IP address class
   e. MAC (not Message Authentication Code)
   f. MAC address
   g. Network Interface Card
   h. Network Address Translation
   i. Private address classes

3) Explain the purpose of the following devices, and at what OSI layer they function:
4) How many IP address classes are there? How many networks and hosts can each address class support?

5) What class does each of the following IP address belong:
   a. 128.238.35.80
   b. 120.238.45.10
   c. 10.0.0.10
   d. 199.219.156.23
   e. 128.138.256.30
   f. 210.10.20.30
   g. 222.111.22.11
   h. 178.251.23.1

6) Consider the following network:

   ![Figure 1](image)

   Assume that all of the machines were just switched on.
   a. Explain how Host A would send a 2048 byte message to Host C using IP (host A, B, and C are in the same subnet and all hosts know each other’s IP address)
   b. Now host A wants to send 2048 byte IP broadcast message to all hosts in its subnet, explain how host A would do the broadcast?

7) What is the difference between UDP and TCP? Explain in terms of primary functions, robustness, reliability, and cost of transmission.

8) What is range of hosts and subnets supported by a class B network using subnet mask 255.255.224.0 List IP range (for example: x.y.32.1 . x.y.63.254) and the network ID for each subnet supported?
9) Logon as root to the node assigned to your group and understand how the following utilities work:
   a. \texttt{arp} utility (type \texttt{man arp} or search google to understand \texttt{arp}) Answer the following questions:
      1. What does \texttt{arp} do?
      2. List the contents of the MAC address cache?
      3. What is the difference between static and dynamic \texttt{arp} cache entry?
      4. How can you bind a MAC address to some IP address in your subnet (statically and dynamically)
      5. List 5 things you can do with \texttt{arp} and how?
   b. \texttt{ifconfig} utility (type \texttt{man ifconfig} or search google for \texttt{ifconfig}) Answer the following questions:
      1. What is the primary purpose of \texttt{ifconfig} utility?
      2. Explain what you see when you type \texttt{ifconfig} in the command line. You have to give an explanation for each entry.
      3. What is your host MAC and IP address?
      4. List 5 things you can do with \texttt{ifconfig} and how?
   c. \texttt{tcpdump} utility (type \texttt{man tcpdump} or search google for \texttt{tcpdump}) Answer the following questions:
      1. What is the primary purpose of \texttt{tcpdump} utility?
      2. Type \texttt{tcpdump > tcpd.txt}, press \texttt{ctrl + c} after 10 seconds. Open tcpd.txt in an editor and explain its contents.
      3. List 5 things you can do with \texttt{tcpdump} and how?
      4. List 5 sites where you find tools to analyze \texttt{tcpdump} outputs (open source tools only).
   d. \texttt{ping} utility (type \texttt{man ping} or search google for \texttt{ping}) Answer the following questions:
      1. What is the primary purpose of \texttt{ping} utility?
      2. Do the following experiment:
         1. Type \texttt{tcpdump > pingexp.txt}
         2. Open a new session
         3. Clear your arp cache
         4. Subtract 1 from the last octet of your IP address and ping the resulting IP address.
         5. Press \texttt{ctrl+c} in the session that is running \texttt{tcpdump}
         6. Open pingexp.txt and explain how ping works by analyzing the contents of pingexp.txt.
      3. List 5 possible things you can do with \texttt{ping}.
      4. List 5 sites where you can find hints about things that you can do with \texttt{ping} utility.

Part II: TCP/IP Routing
Routing is one of the reason TCP/IP has become the core protocol of the Internet. Study how routing works by answering the following questions:
1) What is the difference between packet, circuit, and cell switched networks? Can TCP/IP work in all three networks?

2) What are the contents of an IP header? Draw labeled diagram of a typical IP packet and summarize its contents.

3) Let’s assume that host A with IP 128.238.35.20 and subnet mask 255.255.224.0 wants to send a packet to host B with IP 128.238.160.10 using the same subnet mask as host A. Find if host B is in the same subnet as host A and explain how host A determines if the destination is in its own subnet or not? (Explain how it is done in TCP/IP, and not how you think it should work)

4) What does “IP routing a packet” mean? Give an example TCP/IP network where routing is necessary. Show how data is routed in your network.

5) Consider the following network:

![Diagram of network with Host A, Device S, and Host B]

If device S is a router, then the IP address for each of its interface is the first IP address of the subnet it's facing. For example, the IP address for the interface facing network A would be x.y.z.1

x in figure 2 could take on any of the following values:

a. 8
b. 12
c. 16
d. 20
e. 24
f. 28

a) What is device S for each value of x (i.e. what is device S is if x takes 8 and then 12 and so on), and explain why? If the Device S is router then list the IP address for its interfaces.

b) Host A wants to send a packet to host B (Assuming host A knows host B’s IP address). Explain how this transaction would take place by listing all packets sent by host A, device S, host B. You should also say what each packet is for (i.e. ARP discovery packet, ARP reply… etc) for each value of x. (Everything is just turned on, no entries in arp cache, and device S has static configuration)

6) Summaries the purpose and differences of the following routing protocols:

a. RIP
b. OSPF
c. BGP
7) Logon as root to a node assigned to your group and understand how the following utilities work:

   e. traceroute utility (type `man traceroute` or search google to understand traceroute) Answer the following questions:
      1. What does traceroute do?
      2. List 5 things you can do with traceroute and how?

   f. nmap utility (type `man nmap` or search google for nmap) Answer the following questions:
      1. What is the primary purpose of nmap utility?
      2. List 5 things you can do with nmap and how?

Part III: Network Layout Discovery

Logon as root to a node assigned to your group and do the following exercise:

Imagine that you have been asked to be the administrator for an enterprise network. As a good network administrator it is important for you know the layout of the network. Network layout would help you to better understand the boundaries of your network and it is the knowledge you need to have to diagnose and isolate security problems in your network. Therefore, gathering information about network layout is the first step in securing a network.

Usually network layouts and the components that make a network are documented, but in many cases the documentation is soon outdated due to the dynamics of the network and lack of interest in keeping the document up to date. In this part you will be dealing with such a situation, in which you will have to discover the layout of a given network using some basic tools.

Part IIIa: Node Discovery

With out discovering the nodes in a network it’s not possible to discover its layout. Use the following tools to discover nodes in your network:

1) `tcpdump`: `tcpdump` is the simplest, but very useful tool to discover nodes in a network. The drawback with `tcpdump` is: in a switched environment `tcpdump` can only see broadcast packets from nodes in your network segment and packets designated to you, why? Use `tcpdump -e -x` to capture the packets, and save them in a file. You must let `tcpdump` to capture at least for a minute or so. From the output you should be able to find the IP address of all the networking devices that are communicating to you or broadcasting in your segment.

2) `nmap`: Since `tcpdump` cannot identify silent nodes, use `nmap` to discover all nodes in 10.11.0.0/16 network.

Part IIIb: Network Mapping:

Try to map the network from the IP addresses you found in Part IIIa. You should be able to trace the route from each host, if the host is in the same segment you will get the reply from the host directly, in which case you will have only one entry. If the host is in another segment then you will have more than one reply indicating the routers along
the way to the destination from your host. You can trace an IP address using traceroute or ping with record route (IP RR) option.

**What to Submit for Part III**
Submit a report for PART III that includes following:
- One page of summary on traceroute outputs and a diagram showing the layout of the network that you have discovered.

**Part VI: Linux Socket Programming**

1) Consider the following server code:
```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

static void die(const char *str) {
    printf("%s \n", str);
    exit(1);
}

int main() {
    int state;
    char *nodeAddr;
    char *nodePort;
    struct sockaddr_in nodeSkt;
    struct sockaddr_in clntSkt;
    int len_inet;
    int skt;
    int Clntsocket;
    int bytes;
    char data_buffer[128];

    nodeAddr = "192.168.1.104";
    nodePort = "80"; //Default port
    skt = socket(PF_INET,SOCK_STREAM,0);
    if ( skt == -1 )
        die("Some thing wrong with Socket()");
    memset(&nodeSkt,0,sizeof (nodeSkt));
    nodeSkt.sin_family = AF_INET;
    nodeSkt.sin_port = htons(atoi(nodePort));
    nodeSkt.sin_addr.s_addr = inet_addr(nodeAddr);
    if ( nodeSkt.sin_addr.s_addr == INADDR_NONE )
        die("You typed your node address incorrectly");
    len_inet = sizeof(nodeSkt);
    state = bind(skt,(struct sockaddr *)&nodeSkt,len_inet);
```
if ( state == -1 )
    die("I could not bind()!");
state = listen(skt,10);

if ( state == -1 )
    die("I'm deaf! :( Look what you did to me");
for (;;) {
    len_inet = sizeof(clntSkt);
    Clntsocket = accept(skt, (struct sockaddr *)&clntSkt,
    &len_inet);
    printf("Accepted Client\n");
    if ( Clntsocket == -1 )
        die("I'm too picky0");

    strcpy(data_buffer, "Hi there! you did it");
    bytes = strlen(data_buffer);
    state = write(Clntsocket,data_buffer,bytes);
    if ( state == -1 )
        die("I'm fine, but the other guy/gal messed up. Or
    may be its me");
    close(Clntsocket);
}
return 0; //Will this code ever reach this point?
}

You can compile the above code and run it in the node assigned to you as
administrator. From any workstation in the test bed you should be able to telnet to
port 80 of your node to see if this code work.

Do the following exercise:

1) Explain the above code step by step, that is, explain what each system
call does in the code.

2) Modify the code to display “Hi <client IP> there. You are client
number <i>” when a client telnet to port 80 of your node. <client IP>
is the IP address of the client, and <i> is running counter that is
incremented each time a new client is accepted.