SPL-1 Project Report, 2019

Port Scanner

SE 305 : Software Project Lab-1

Submitted by

MD Akram Khan
BSSE Roll No. : 1007
BSSE Session: 2017-2018

Supervised by

Dr. B M Mainul Hossain
Designation: Associate professor
Institute of Information Technology

Institute of Information Technology,
University of Dhaka
29-05-2019

Table of Contents:

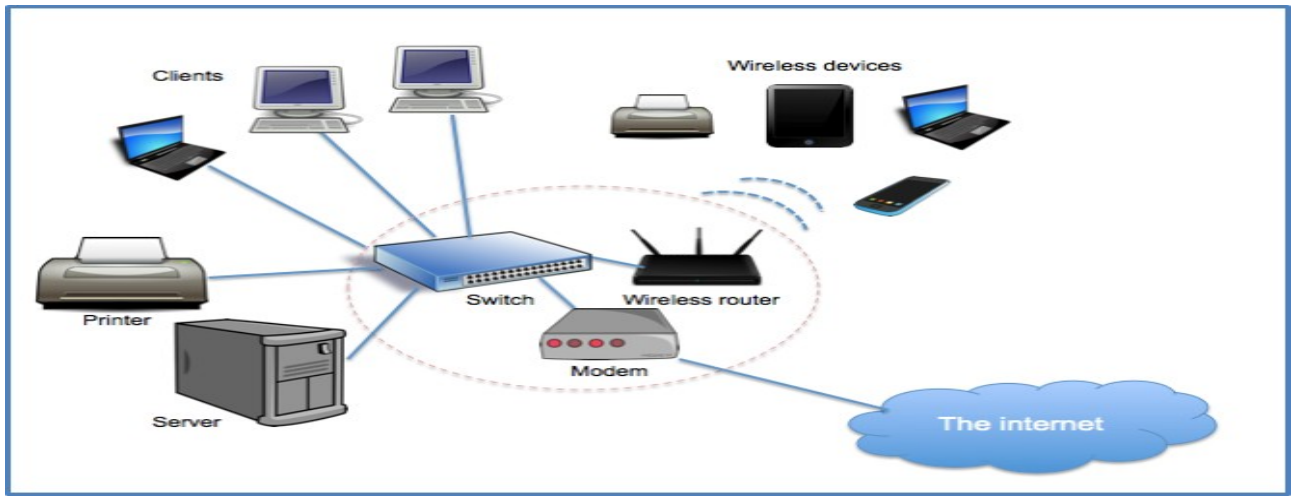
1.Introduction

A port scan is a method for determining which ports on a network are open. As ports on a computer are the place where information is sent and received, port scanning is analogous to knocking on doors to see if someone is home.

Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicate whether the port is used and can be probed for further weakness. Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system.

1.1 Background study

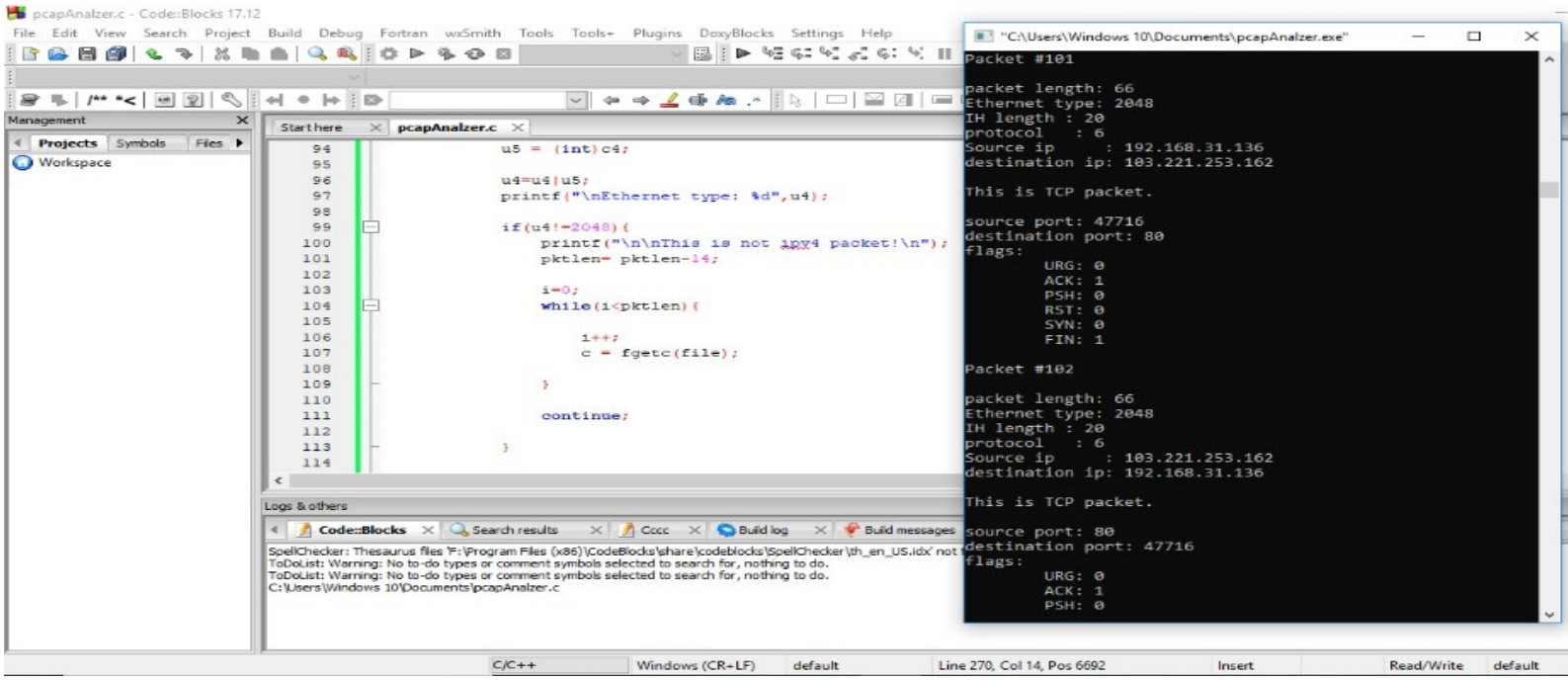
Networking Basics : To make a port scanner , I needed to learn the networking basic topics like OSI layer, protocol etc and keywords like port, firewall, lan etc .



Source : https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols

PCAP File Analyzer : I needed to learn the packet parsing. I have also studied about the TCP, IP header and written a code for packet

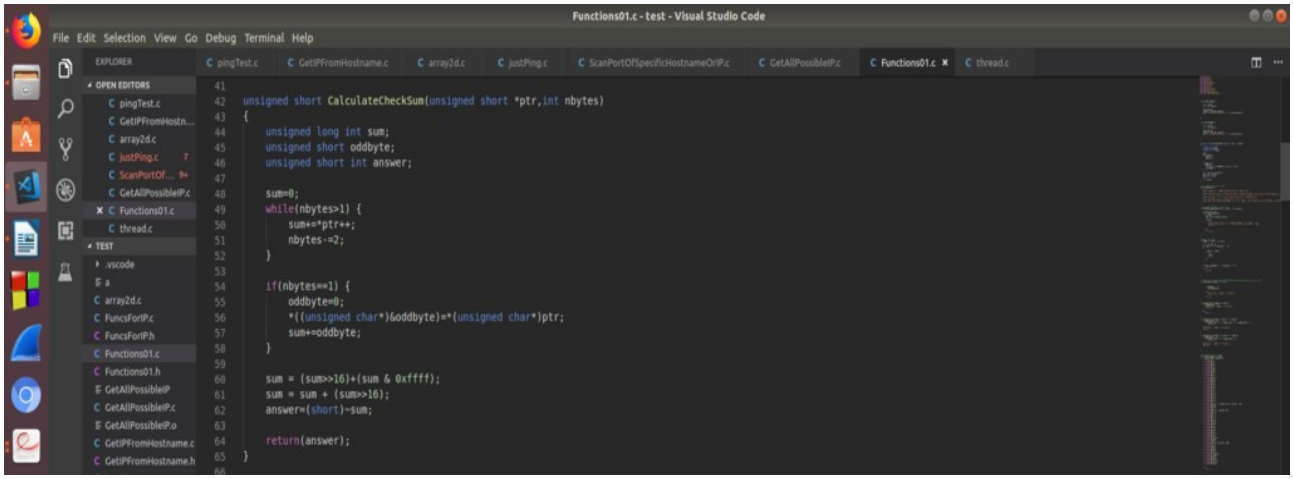
parsing from a pcap file so that I can find out the source IP address , destination IP address, source and destination port no , sequence number , acknowledgement number of any packet from this pcap file.



Raw Socket : A raw socket is a type of socket that allows access to the underlying transport provider. I have learned raw socket to send tcp packet to specific host to scan port of this host.

Checksum Calculation : A checksum is an error-detection method. To test data integrity, the sender of the data calculates checksum value by taking the sum of the binary data transmitted. When receiving the data,

the receiver can perform the same calculation on the data and compare it with the checksum value provided by the sender. If the two values match, the receiver has a high degree of confidence that the data was received correctly.



1.2. Challenges

Implementing a new software solution carries with it a number of challenges. The process can be overwhelming , confusing and lengthy. For implementing this project there are lot of challenges that I have faced. Some of them are :

1) Using multiple file for c source code , which was difficult for me.

2) Sending packet using raw socket properly.

3) Receiving the reply packet from any host correctly.

4) Understanding the state of any port of specific host.

5) Using timer for filtered port of any host.

6) Facing problem for not understanding the term endianness .

2. Project Overview

TCP SYN scan is used in this port scanner to scan the ports.

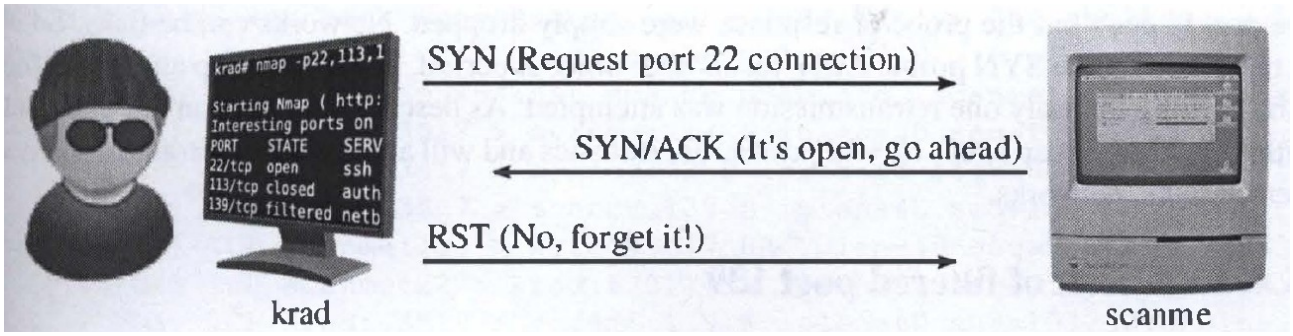
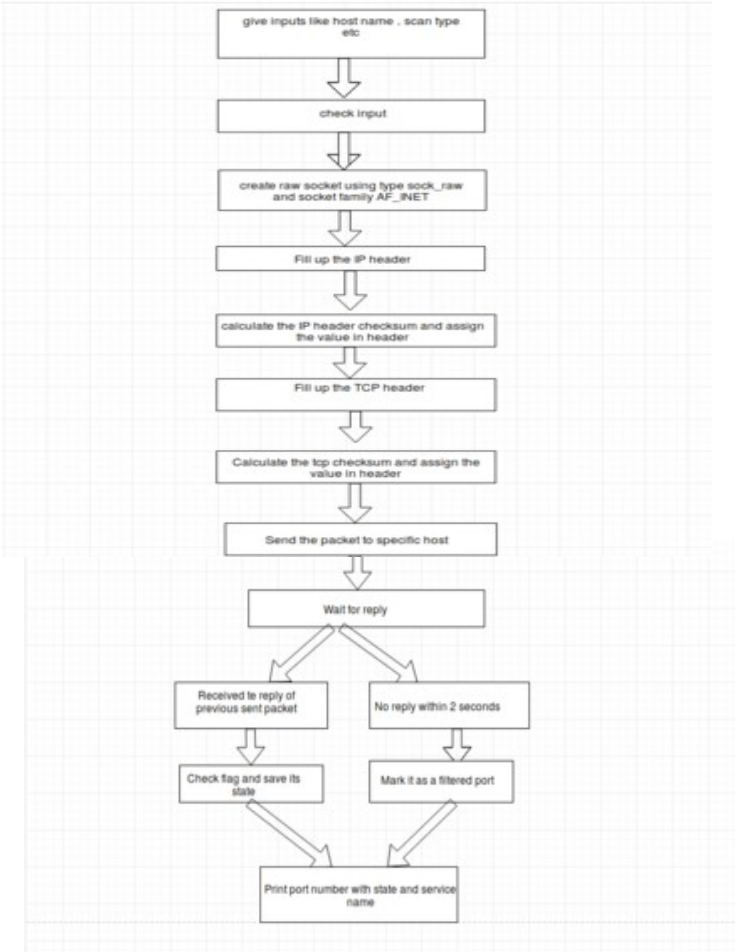


image source : https://medium.com/@avirj/nmap-tcp-syn-scan-50106f818bf1



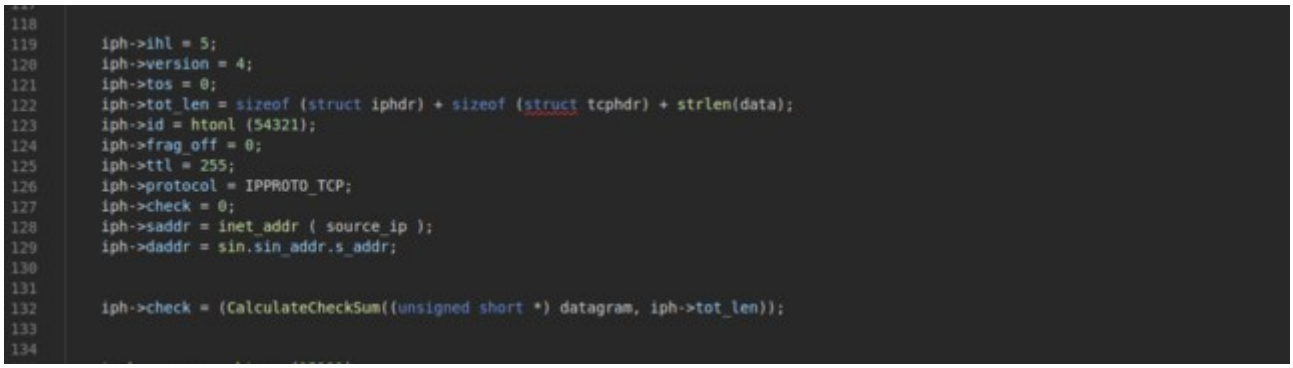
At first, inputs given by the user will be checked whether it is correct or not .

```
int isGivenThreeCorrectInput(char argv2[]){
    if(!isEqual(argv2,"-P")){
        printf("Type \"./akMap -h \"for Help\n");
        return 0;
    }
    else
        return 1;

}

int isGivenFourCorrectInput(char argv2[], char argv3[]){
    if(isEqual(argv2,"-S"))
        if(isEqual(argv3,"-A") || isEqual(argv3,"-D") || isEqual(argv3,"-F") )
            return 1;

    printf("Type \"./akMap -h \"for Help\n");
    return 0;

}
```

Using this Port Scanner , can detect the state of specific port of any live host . Raw socket is used to send and receive the TCP packet.

```
    SOCK = socket (AF_INET, SOCK_RAW, IPPROTO_TCP);

    if(SOCK == -1)
    {
        perror("FAILED to create SOCKET!");
        exit(1);
    }
```

Then the IP header is filled up properly as we have used raw used. Here checksum is calculated after the filling up all the value so that the checksum remains correct.



```
    iph->ihl = 5;
    iph->version = 4;
    iph->tos = 0;
    iph->tot_len = sizeof (struct iphdr) + sizeof (struct tcphdr) + strlen(data);
    iph->id = htonl (54321);
    iph->frag_off = 0;
    iph->ttl = 255;
    iph->protocol = IPPROTO_TCP;
    iph->check = 0;
    iph->saddr = inet_addr ( source_ip );
    iph->daddr = sin.sin_addr.s_addr;


    iph->check = (CalculateCheckSum((unsigned short *) datagram, iph->tot_len));
```

Then the TCP header is filled up properly. Here checksum is calculated after the filling up all the value and filling up the pseudo header so that the checksum remains correct.

```
tcph->source = htons (15111);
tcph->dest = htons (portNo);
tcph->seq = 110;
tcph->ack_seq = 110;
tcph->doff = 5;
tcph->fin=0;
tcph->syn=1;
tcph->rst=0;
tcph->psh=0;
tcph->ack=0;
tcph->urg=0;
tcph->window = htons (5840);
tcph->check = 0;
tcph->urg_ptr = 0;


psh.source_address = inet_addr( source_ip );
psh.dest_address = sin.sin_addr.s_addr;
psh.placeholder = 0;
psh.protocol = IPPROTO_TCP;
psh.tcp_length = htons(sizeof(struct tcphdr) + strlen(data) );

psize = sizeof(struct pseudo_header) + sizeof(struct tcphdr) + strlen(data);
pseudogram = malloc(psize);

memcpy(pseudogram , (char*) &psh , sizeof (struct pseudo_header));
memcpy(pseudogram + sizeof(struct pseudo_header) , tcph , sizeof(struct tcphdr) + strlen(data));

tcph->check = CalculateCheckSum( (unsigned short*) pseudogram , psize);
```

Instead of computing the checksum over only the actual data fields of the TCP segment , a 12-byte TCP *pseudo header* is created prior to checksum calculation. It contains the most important parts of the IP header, that is, source and destination address, protocol number and data length.

```
27
28    struct pseudo_header
29    {
30        unsigned int source_address;
31        unsigned int dest_address;
32        unsigned char placeholder;
33        unsigned char protocol;
34        unsigned short int tcp_length;
35    };
36
```

Checksum is used to ensure the integrity of a file after it has been transmitted from one storage device to another. This can be across the Internet or simply between two computers on the same network.

```
unsigned short CalculateCheckSum(unsigned short *ptr,int nbytes)
{
    unsigned long int sum;
    unsigned short oddbyte;
    unsigned short int answer;

    sum=0;
    while(nbytes>1) {
        sum+=*ptr++;
        nbytes-=2;
    }

    if(nbytes==1) {
        oddbyte=0;
        *((unsigned char*)&oddbyte)=*(unsigned char*)ptr;
        sum+=oddbyte;
    }

    sum = (sum>>16)+(sum & 0xffff);
    sum = sum + (sum>>16);
    answer=(short)~sum;

    return(answer);
}
```

Now, the packet needs to be sent to the specific port of specific host.

```
if (sendto (SOCK, datagram, iph->tot_len ,  0, (struct sockaddr *) &sin, sizeof (sin)) < 0)
{
    perror("sendto failed");
    exit(0);

}
```

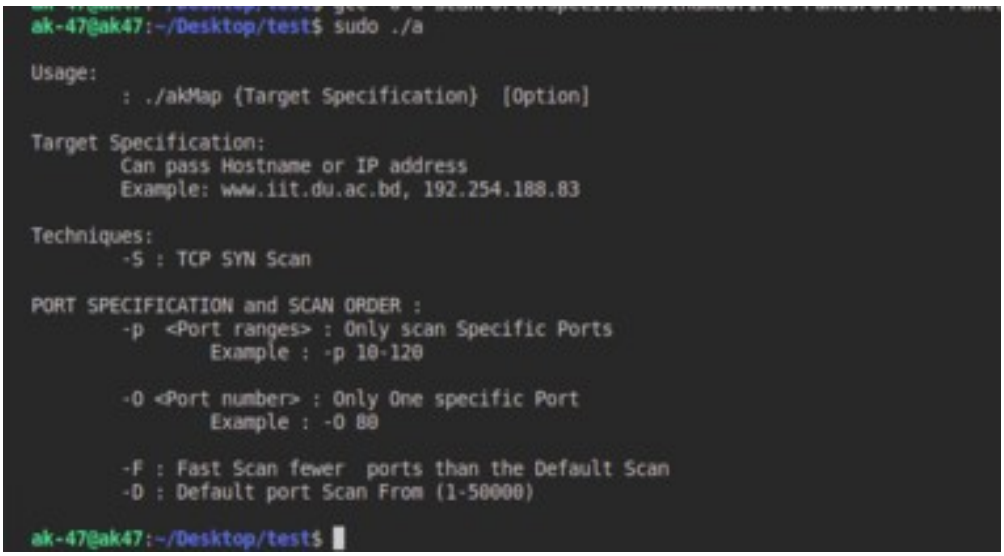
If the host machine is live and it doesn't have any firewall, then it will receive the packet and , it will reply with another packet . From the reply, the poer state can be understood. If in the reply packet both RST and ACK flag value is 1, then this port can be said to be closed. Again if in the reply packet the ACK and SYN flag value is 1 , then the port can be claimed as open. If there is no reply within 2 minutes , then the port will be considered as filtered.

```
217                     portState[counter] = "open";
218                     openNfilteredPort[counter] = portNo;
219                     counter++;
220                     break;
221
222                 }
223
224
225                 else if(tcp->rst && tcp->ack){
226                     closedPortCounter++;
227                     break;
228                 }
229
230             }
231
232         }
233
234     }
235
236     if(flag){
237
238         portState[counter]="filtered";
239         openNfilteredPort[counter] = portNo;
240         counter++;
```

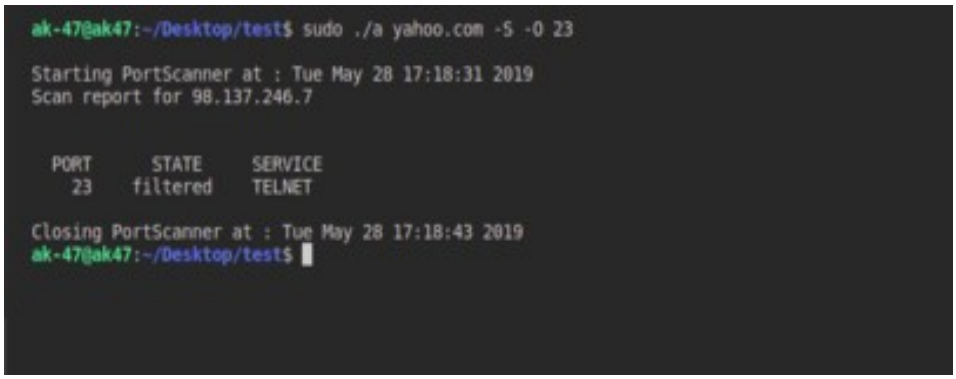
```
ak-47@ak47:~/Desktop/test$ sudo ./a yahoo.com -S -O 23

Starting PortScanner at : Tue May 28 17:18:31 2019
Scan report for 98.137.246.7


  PORT      STATE     SERVICE
    23    filtered    TELNET

Closing PortScanner at : Tue May 28 17:18:43 2019
ak-47@ak47:~/Desktop/test$
```

3. User manual

First go to the ubuntu terminal and Browse to the folder where the executable file is stored. Then, type the command **“ sudo ./executable fileName -h ”** to see the user manual.

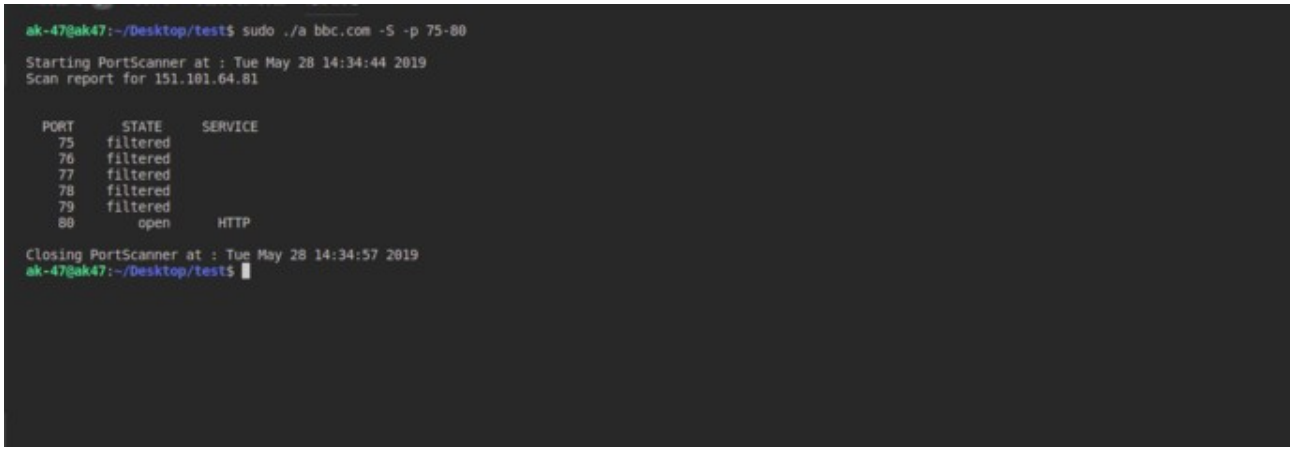


1) Only one random port of specific host can be scanned using the following command:
“sudo ./executableFileName hostName -S -O portNo ”



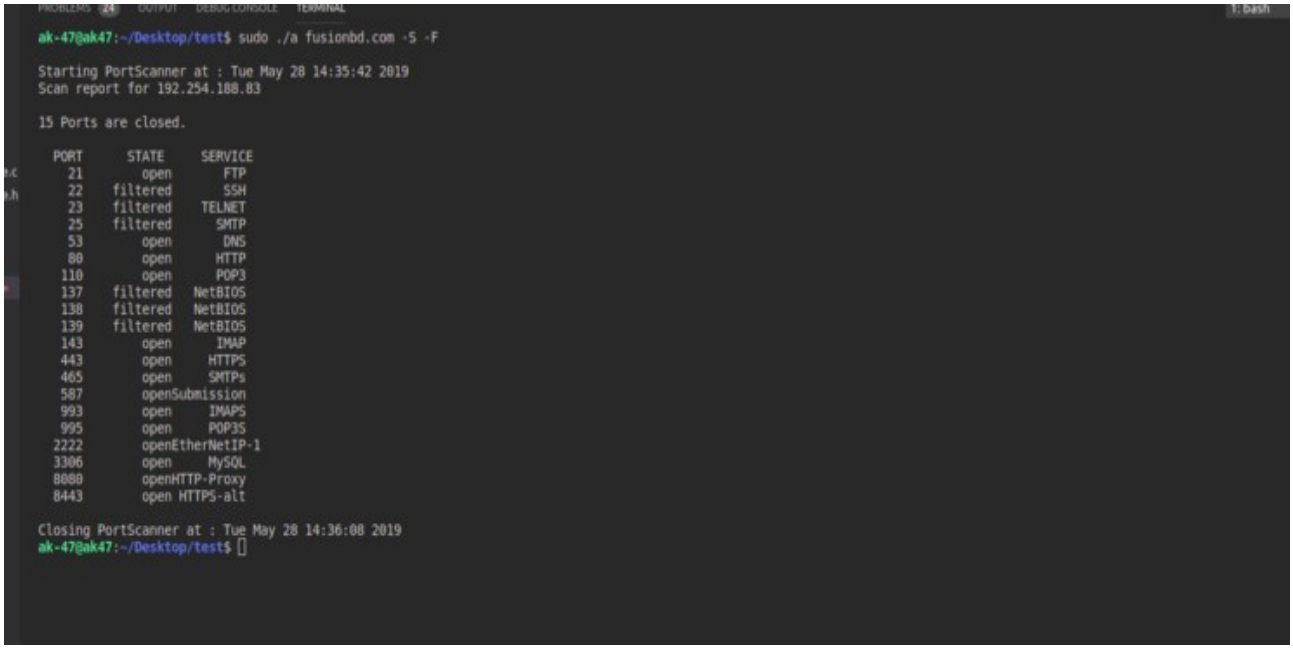
2) More than one port can also be scanned using this port scnner using the following command :

“sudo ./executableFileName hostName -S -p startingPortNo -endingPortNo ”



3) Some common ports of specific host can be scanned using this scanner by the following command :

“sudo ./executableFileName hostName -S -F”



4) Default scan is used to scan ports from 1-50000 sequentially using the following command :

“sudo ./executableFileName hostName -S -D”

4. Conclusion

This project helps me to understand computer networking and improve my coding skill and I have learned to handle multiple source file in c for the first time. I hope it will help me to deal with difficulties in future. This project was quiet challenging as I didn't have any knowledge about computer networking before doing this project and I gained a lot of experience from it. Hope this knowledge will help me in my future networking related projects. I want to thank my supervisor for guiding me a lot during this project.

5. Appendix

In this project, I have made a port scanner . I have also tried to send icmp packet to any host to check whether it is live or not, but, I am facing so many difficulties in doing this. In future, I have plan to complete this task, so that I can scan all the host in my LAN to check , which machines are live . I will also be able to ping any valid host to see whether the server machine is live or not.

6. Reference

1) PCAP format , https://github.com/hokiespurs/velodyne-copter/wiki/PCAP-format, [26-01-2019]

2) Pcap Headers Description , https://github.com/hokiespurs/velodyne-copter/wiki/PCAP-format[27-01-2019]

3) Global Header , https://wiki.wireshark.org/Development/LibpcapFileFormat#Global_Header, [28-01-2019]

4) checksum calculation , https://www.winpcap.org/pipermail/winpcap-users/2007-July/001984.html , [25-02-2019]

5) Ipv4 header format , https://www.gatevidyalay.com/ipv4-ipv4-header-ipv4-header-format/ , [02-4-2019]

6) exploring data packet , https://www.techrepublic.com/article/exploring-the-anatomy-of-a-data-packet/ , [01-04-2019]

7) Ipv4 header format , https://community.cisco.com/t5/other-network-architecture/ip-header-ihl-field-what-is-the-maximum-ip-header-length/td-p/197444 , [02-4-2019]

8) Structures defined in the sys/socket.h header file, https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.14/gtpc1/gtpc1mst118.html, [10-04-2019]

9) TCP Header Format, https://www.freesoft.org/CIE/Course/Section4/8.htm, [04-04-2019]

10) SYN scanning , https://searchnetworking.techtarget.com/definition/SYN-scanning, [18-01-2019]