Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Department of Cyber Security

Laboratory Manual

Academic Year:2023-24

Name of the Course: **Computer Networks Lab**

Course Code :22CY2252

Regulation:R22

Year & Semester: II-B.Tech& II-Semester

Prepared By

P.Chandra Mounika

Assistant Professor

DEPARTMENT VISION & MISSION

Department Vision

To be a center of excellence through innovative education, research, and collaborative partnership, with strong ethics and adaptability to raise cyber awareness, build robust defense, and foster a cyber-resilient ecosystem that safeguards critical information and digital assets.

Department Mission

DM-1: To enhance the knowledge and skills of the students through effective teaching and learning processes to become cybersecurity professionals with the technical expertise and passion needed to protect digital assets.

DM-2: To familiarize students with the latest tools and methodologies to combat cyber threats and identify and mitigate cyber-attacks in cyberspace.

DM-3: To inculcate the habit of attaining firm ethical grounding by conforming to IT Laws and nurturing research abilities.

Program Outcomes:
Engineering graduates will be able to:

PO1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, and environmental considerations.
PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective Presentations, and give and receive clear instructions.

PO 11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.
PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO's)

PSO-1: Demonstrate expertise in ethical hacking techniques, proficiency in identifying vulnerabilities and potential threats in systems and networks.

PSO-2: Showcase mastery in collecting, preserving, and analyzing digital evidence to support legal investigations and cybercrime cases.

PSO-3: Exhibit managerial skills and knowledge along with certification in ethical hacking which is globally recognized to predict future challenges in protecting digital assets of individuals, organizations and society.

Course Objectives:

To understand the working principle of various communication protocols.

To understand the network simulator environment and visualize a network topology and observe its performance.

To analyze the traffic flow and the contents of protocol frames

S.NO	Course out comes (CO's)	Taxonomy level
CO1	To understand the working principle of various communication protocols.	**L2**
CO2	To understand the network simulator environment visualize a network topology and observe its performance.	**L2**
CO3	To understand and visualize a network topology and observe its performance.	**L2**
CO4	To analyze the traffic flow and the contents of protocol frames	**L3**
CO5	To analyze the contents of protocol frames	**L3**

Course Outcomes:

Implement data link layer farming methods

Analyze error detection and error correction codes.

Implement and analyze routing and congestion issues in network design.

Implement Encoding and Decoding techniques used in presentation layer

To be able to work with different network tools

Course Outcome	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	2	2	2	1	3								1		
CO2	2	2	2		3										
CO3	2	2	3	1	3										
CO4	2	2	3		3				3						
CO5	2	2	3		2				3				1		
TOTAL	**1.5**	**2**	**2.6**	**1**	**2.6**	**-**		**-**	**3**			**-**			

1. GUIDELINES TO STUDENTS

1.Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.

2.Students are instructed to come to lab in formal dresses only.

3.Students are supposed to occupy the systems allotted to them and are not supposed to talk or make noise in the lab.

4.Students are required to carry their observation book and lab records with completed exercises while entering the lab.

5.Lab records need to be submitted every week.

6.Students are not supposed to use pen drives in the lab.

List of experiments as per the curriculum

S.No.	Name of the Program
1	**Week1 :**
	Implement the data link layer framing methods such as character, character stuffing and bit stuffing.
2	**Week 2 :**
	Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRCCCIP
3	**Week 3 :**
	Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4.	**Week 4 :**
	Implement Dijkstra's algorithm to compute the shortest path through a network
5	**Week 5 :**
	Take an example subnet of hosts and obtain a broadcast tree for the subnet
6	**Week 6 :**
	Implement distance vector routing algorithm for obtaining routing tables at each node
7	**Week 7 :**
	Implement data encryption and data decryption
8	**Week 8 :**
	Write a program for congestion control using Leaky bucket algorithm
9	**Week 9 :**
	Write a program for frame sorting techniques used in buffers
10	**Week 10 :**
	Wireshark i. Packet Capture Using Wire shark ii. Starting Wire shark iii. Viewing Captured Traffic iv. Analysis and Statistics & Filters.
11	**Week 11 :**
	How to run Nmap scan
12	**Week 12 :**
	Operating System Detection using Nmap
13	**Week 13 :**
	Do the following using NS2 Simulator i. NS2 Simulator-Introduction ii. Simulate to Find the Number of Packets Dropped iii. Simulate to Find the Number of Packets Dropped by TCP/UDP iv. Simulate to Find the Number of Packets Dropped due to Congestion v. Simulate to Compare Data Rate & Throughput. vi. Simulate to Plot Congestion for Different Source/Destination vii. Simulate to Determine the Performance with respect to Transmission of Packet

1. Implement the data link layer framing methods such as character, character stuffing and bit stuffing.

Character Count:

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
   char data[30],stuff_data[40];
   int i,j,dsize,fsize,totalframes,tframe,s;
printf("Enter Data:");
scanf("%s",data);
dsize=strlen(data);
printf("Enter Frame Size:");
scanf("%d",&fsize);
tframe=fsize-1;
totalframes = (dsize/tframe) + 1;
   s=fsize;
   j=0;
   for(i=0;i<(dsize+totalframes);i++)
   {
      if(i%fsize==0)
      {
         if((dsize-j)<fsize)
         {
fsize=dsize-j+1;
         }
stuff_data[i]=(char)(48+fsize);
fsize=s;
      }
      else
      {
stuff_data[i]=data[j];
j++;
      }
   }
stuff_data[dsize+totalframes+1]='\0';
printf("Stuffed data is :%s",stuff_data);
}
```

OUTPUT-1:

```
Enter Data:cybersecurity
Enter Frame Size:5
Stuffed data is :5cybe5rsec5urit2y
```

OUTPUT-2:

```
Enter Data:hihellowelcome
Enter Frame Size:4
```

```
Stuffed data is :4hih4ell4owe4lco3me
```

BIT STUFFING:

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
   char data[30],stuff_data[40],delimiter[10]="01111110";
   int i,j,count,len;
printf("Enter Data:");
scanf("%s",data);
len=strlen(data);
   count=0;
   j=0;
   for(i=0;i<len;i++)
   {
      if(data[i]=='1')
      {
         count++;
      }
      else
      {
         count=0;
      }
stuff_data[j]=data[i];
j++;
if(count==5 && data[i+1]=='1')
      {
stuff_data[j]='0';
j++;
         count=0;
      }
   }
stuff_data[j]='\0';
strcat(stuff_data,delimiter);
strcat(delimiter,stuff_data);
strcpy(stuff_data,delimiter);
printf("\nStuffed data is :%s",stuff_data);
}
```

OUTPUT-1:

```
Enter Data:1011110
Stuffed data is :0111111010111101111110
```

OUTPUT-2:

```
Enter Data:01011111101
Stuffed data is :01111110010111110101 01111110
```

CHARACTER STUFFING:

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
   int i,j,count=0,l,p;
   char a[30],b[4]="DLE";
printf("enter the data frame :");
scanf("%s",a);
   l=strlen(a);
printf("DLESTX ");
   for(i=0;i<l;i++)
   {
      count=0;
      p=i;
      for(j=0;j<3;j++)
      {
         if(a[i]==b[j])
         {
            count=count+1;
i++;
         }
      }
      if(count!=3)
      {
i=p;
      }
      if(count==3)
      {
printf("DLEDLE");
      }
      else
      {
printf("%c",a[i]);
      }
   }
printf(" DLEETX");
}
```

OUTPUT-1:

```
enter the data frame :VBIT
DLESTX VBIT DLEETX
```

OUTPUT-2:

```
enter the data frame :DOODLE
DLESTX DOODLEDLE DLEETX
```

2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRCCCIP.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
   char poly[20],frame[50],stuff_data[50];
   int i,j,k,polylen,framelen,sdlen;
printf("Enter Frame:");
scanf("%s",frame);
printf("Enter Polynomial:");
scanf("%s",poly);
polylen=strlen(poly);
framelen=strlen(frame);
   for(i=0;i<polylen;i++)
   {
      if(poly[i]=='1')
      {
polylen=polylen-i;
         break;
      }
   }
   for(k=0;k<polylen;k++)
   poly[k]=poly[k+i];
sdlen=polylen+framelen-1;
   for(i=0;i<framelen;i++)
stuff_data[i]=frame[i];
   for(i=framelen;i<sdlen;i++)
stuff_data[i]='0';
stuff_data[i]='\0';
   for(i=0;i<framelen;i++)
   if(stuff_data[i]=='1')
   {
      for(j=i,k=0;k<polylen;k++,j++)
      if(poly[k]==stuff_data[j])
stuff_data[j]='0';
      else
stuff_data[j]='1';
   }
   for(i=0;i<framelen;i++)
stuff_data[i]=frame[i];
printf("the message is:%s",stuff_data);
}
```

OUTPUT:

```
Enter Frame:1010
Enter Polynomial:1001
the message is:1010011
```

3. Develop a simple data link layer that performs the flow control using the sliding window protocol and loss recovery using the Go-Back-N mechanism.

PROGRAM:

```
#include <stdio.h>
#include<stdlib.h>
#include<math.h>
int n,r;
struct frame
{
    char ack;
    int data;
}frame[10];
int sender();
void recack();
void resend();
int main()
{
sender();
recack();
resend();
    return 0;
}
int sender()
{
    int i;
printf("Enter the no.of frames to be sent");
scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
printf("\nEnterdta for frames[%d]:",i);
scanf("%d",&frame[i].data);
        frame[i].ack='y';
    }
return 0;
}
void recack()
{
    int i;
rand();
    r=rand()%n;
    frame[r].ack='n';
    for(i=1;i<=n;i++)
    {
        if(frame[i].ack=='n')
printf("\n The frame number %d is not received",r);
    }
}
 void resend()
 {
     int i;
printf("\n resending from frame %d",r);
     for(i=r;i<=n;i++)
    {
sleep(2);
        frame[i].ack='y';
printf("\n Received data of frame %d is %d",i,frame[i].data);
    }
 }
```

OUTPUT:

```
Enter the no.of frames to be sent4
Enter data for frames[1]:10
Enter data for frames[2]:20
Enter data for frames[3]:30
Enter data for frames[4]:40
 The frame number 2 is not received
 resending from frame 2
 Received data of frame 2 is 20
 Received data of frame 3 is 30
 Received data of frame 4 is 40
```

4. Implement Dijsktra’s algorithm to compute the shortest path through a network.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#include<math.h>
int main()
{
        int u,v,num,i,j,s[10],min,cost[10][10],dist[10],path[10],n;
        printf("\n ENTER VERTECES:");
        scanf("%d",&n);
        printf("\n ENTER ADJECENCY MATRIX(if no edge enter -1):\n");
        for(i=1;i<=n;i++)
        {
                for(j=1;j<=n;j++)
                scanf("%d",&cost[i][j]);
        }
        for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        if(i==j) cost[i][j]=0;
        else if(cost[i][j]==-1)
        cost[i][j]=30000;
        printf("\nENTER SOURCE VERTEX:");
        scanf("%d",&v);
        for(i=1;i<=n;i++)
        {
                s[i]=0;
                path[i]=v;
                dist[i]=cost[v][i];
        }
        dist[v]=0;
        for(num=2;num<=n;num++)
        {
                min=30000; u=0;
                for(i=1;i<=n;i++)
                {
                        if(s[i]!=1)
                        if(min>dist[i])
                        {
                                u=i;
                                min=dist[i];
                        }
                }
                s[u]=1;
                for(i=1;i<=n;i++)
                {
                        if(s[i]!=1)
                        if(dist[i]>(min+cost[u][i]))
                        {
                                dist[i]=min+cost[u][i];
                                path[i]=u;
                        }
                }
        }
        printf("\n");
        printf("\nPATH MATRIX:\n");
        printf("\nDISTANCE PATH  FROM NODE\n");
        for(i=1;i<=n;i++)
        {
                printf("\n %d",dist[i]);
                printf(" %d ",i);
                j=i;
                do
                {
                        printf(" <-- %d",path[j]);
                        u=path[j];
                        j=u;
                }while(u!=v);

        }
}
```

OUTPUT:

```
ENTER VERTECES:4

 ENTER ADJECENCY MATRIX(if no edge enter -1):
0 3 2 5
3 0 -1 1
2 -1 0 4
5 1 4 0

ENTER SOURCE VERTEX:1


PATH MATRIX:
```

```
DISTANCE PATH  FROM NODE

 0 1  <-- 1
 3 2  <-- 1
 2 3  <-- 1
 4 4  <-- 2 <-- 1
```

5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.

PROGRAM:

```
#include<stdio.h>
int a[10][10],n;
void adj(int);
int main()
{
        int i,j,root;
        printf("Enter no.of nodes:");
        scanf("%d",&n);
        printf("Enter adjacent matrix(in 0’s and 1’s)\n");
        for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
                printf("Enter connecting of %d-->%d::",i,j);
                scanf("%d",&a[i][j]);
        }
        printf("Enter root node:");
        scanf("%d",&root);
        adj(root);
}
void adj(int k)
{
        int i,j;
        printf("Adjacent nodes of root node %d::\n",k);
        for(j=1;j<=n;j++)
        {
                if(a[k][j]==1 || a[j][k]==1)
                printf("%d\t",j);
        }
        printf("\n");
        printf("Nonadjacent nodes of root node %d are ::\n",k);
        for(i=1;i<=n;i++)
        {
                if((a[k][j]==0) && (a[i][k]==0) && (i!=k))
                printf("%d",i);
        }

}
```

OUTPUT:

```
Enter no.of nodes:4
Enter adjacent matrix(in 0’s and 1’s)
Enter connecting of 1-->1::0
Enter connecting of 1-->2::1
Enter connecting of 1-->3::1
Enter connecting of 1-->4::0
Enter connecting of 2-->1::1
Enter connecting of 2-->2::0
Enter connecting of 2-->3::0
Enter connecting of 2-->4::1
Enter connecting of 3-->1::1
Enter connecting of 3-->2::0
Enter connecting of 3-->3::0
Enter connecting of 3-->4::1
Enter connecting of 4-->1::0
Enter connecting of 4-->2::1
Enter connecting of 4-->3::1
Enter connecting of 4-->4::0
Enter root node:1
Adjacent nodes of root node 1::
2       3
Nonadjacent nodes of root node 1 are ::
4
```

6. Implement distance vector routing algorithm for obtaining routing tables at each node.

PROGRAM:

```
#include<stdio.h>
struct node
{
   unsigned dist[20];
   unsigned from[20];
}rt[10];
int main()
{
   int costmat[20][20];
   int nodes,i,j,k,count=0;
printf("\nEnter the number of nodes : ");
scanf("%d",&nodes);
printf("\nEnter the cost matrix :\n");
   for(i=0;i<nodes;i++)
   {
      for(j=0;j<nodes;j++)
      {
scanf("%d",&costmat[i][j]);
costmat[i][i]=0;
         rt[i].dist[j]=costmat[i][j];
         rt[i].from[j]=j;
      }
   }
   do
   {
      count=0;
      for(i=0;i<nodes;i++)
      for(j=0;j<nodes;j++)
      for(k=0;k<nodes;k++)
```

```
        if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
        {
            rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
            rt[i].from[j]=k;
            count++;
        }
}while(count!=0);
    for(i=0;i<nodes;i++)
    {
printf("\n\n For router %d\n",i+1);
        for(j=0;j<nodes;j++)
        {
printf("\t\nnode%d via %d Distance %d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
        }
    }
printf("\n\n");
}
```

OUTPUT:

```
Enter the number of nodes : 3

Enter the cost matrix :
0 3 4
3 0 5
4 5 0


 For router 1

node1 via 1 Distance 0
node2 via 2 Distance 3
node3 via 3 Distance 4

 For router 2

node1 via 1 Distance 3
node2 via 2 Distance 0
node3 via 3 Distance 5

 For router 3

node1 via 1 Distance 4
node2 via 2 Distance 5
node3 via 3 Distance 0
```

7. Implement data encryption and data decryption.

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>

long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[10],en[100],i;
char msg[100];
int prime(long int);
void ce();
long int cd(long int);
void encrypt();
void decrypt();
int main()
{
        printf("\n enter first number\n:");
        scanf("%ld",&p);
        flag=prime(p);
        if(flag==0)
        {
                printf("\n wrong input\n");
                exit(1);
        }
        printf("\n enter another prime number\n ");
        scanf("%ld",&q);
        flag=prime(q);
        if(flag==0||p==q)
        {
                printf("\n wrong input\n");
                exit(1);
        }
        printf("\n enter message\n ");
        fflush(stdin);
        scanf("%s",msg);
        for(i=0;msg[i]!='\0';i++)
        m[i]=msg[i];
        n=p*q;
        t=(p-1)*(q-1);
        ce();
        encrypt();
        decrypt();
}

int prime(long int pr)
{
        int i;
        j=sqrt(pr);
        for(i=2;i<=j;i++)
        {
                if(pr%i==0)
                return 0;
        }
        return 1;
```

```
}
void ce()
{
        int k;
        k=0;
        for(i=2;i<t;i++)
        {
                if(t%i==0)
                continue;
                flag=prime(i);
                if(flag==1&&i!=p&&1!=q)
                {
                        e[k]=i;
                        flag=cd(e[k]);
                        if(flag>0)
                        {
                                d[k]=flag;
                                k++;
                        }
                        if(k==99)
                        break;
                }
        }
}
        long int cd(long int x)
        {
                long int k=1;
                while(1)
                {
                        k=k+t;
                        if(k%x==0)
                        return(k/x);
                }
        }
        void encrypt()
        {
                long int pt,ct,key=e[0],k,len;
                i=0;
                len=strlen(msg);
                while(i!=len)
                {
                        pt=m[i];
                        pt=pt-96;
                        k=1;
                        for(j=0;j<key;j++)
                        {
                                k=k*pt;
                                k=k%n;
                        }
                        temp[i]=k;
                        ct=k+967;
                        en[i]=ct;
                        i++;

                }
```

```
                        en[i]=-1;
                        printf("\n the encrypted message is\n");
                        for(i=0;en[i]!=-1;i++)
                        printf("%c",(char)en[i]);

        }
                void decrypt()
          {
                long int pt,ct,key=d[0],k,i=0;
                while(en[i]!=-1)
                {
                        ct=temp[i];
                        k=1;
                        for(j=0;j<key;j++)
                        {
                                k=k*ct;
                                k=k%n;
                        }
                        pt=k+96;
                        m[i]=pt;
                        i++;
                }
                m[i]=-1;
                printf("\n the decrypted message is \n");
                for(i=0;m[i]!=-1;i++)
                printf("%c",(char)m[i]);
         }
```

OUTPUT:

```
command to compile the program: gcc<program name> -lm
command to run :./a.out

enter first prime number :5
enter another prime number : 11
enter message :hiii
the encrypted message is : #???#
the decrypted  message is : hiii
```

8. Write a program for congestion control using Leaky bucket algorithm.

PROGRAM:

```
#include<stdio.h>
int main()
{
        int incoming,outgoing,bucket_size,n,store=0;
        printf("enter bucketsize,outgoingrate and no. of inputs:");
        scanf("%d%d%d",&bucket_size,&outgoing,&n);
        while(n!=0)
        {
                printf("enter the incoming packetsize:");
                scanf("%d", &incoming);
                printf("incoming packet size %d", incoming);
```

```
		if(incoming<=(bucket_size-store))
		{
			store+=incoming;
			printf("bucket buffer %d\n out of %d\n",store,bucket_size);
		}
		else
		{
			printf("dropped %dno. of packets" , incoming-(bucket_size-store));
			printf("bucket buffer size%d out of %d", store,bucket_size);
			store=bucket_size;
		}
		store=store-outgoing;
		printf("after outgoing %d\n packets left out\n of %d in buffer\n",store,bucket_size);
		n--;
	}
}
```

OUTPUT:

```
 enter bucketsize,outgoing rate and no. Of inputs : 60 20 2
enter the incoming packet size : 30
incoming packet size 30  bucket buffer 30 out of 60 in buffer
after outgoing 10
packets left out 60 in buffer
 enter the incoming packet size : 40
incoming packet size 40  bucket buffer 50 out of 60 in buffer
after outgoing 30 packets left out of 60 in buffer
```

9. Write a program for frame sorting technique used in buffers.

PROGRAM:

```
#include <stdio.h>
struct frame
{
	int num;
	char str[20];
};
struct frame a[10];
int n;
void sort(){
int i,j;
struct frame temp;
for (i=0;i<n-1;i++)
for (j=0;j<n-i-1;j++)
if (a[j].num > a[j+1].num){
	temp=a[j];
	a[j]=a[j+1];
	a[j+1]=temp;
}
}
int main(){
int i ;
printf("Enter no of frames :");
scanf("%d",&n);
printf("Enter frame numbers and its content");
for(i=0;i<n;i++)
scanf("%d %s",&a[i].num,&a[i].str);
sort();
printf("The frame in sequence \n");
for (i=0;i<n;i++)
printf("\n %d \t %s \n ",a[i].num,a[i].str);
}
```

OUTPUT:

```
Enter no of frames :5
Enter frame numbers and its content2b
5e
1a
4d
3c
The frame in sequence

 1       a

 2       b

 3       c

 4       d

 5       e
```

10. Wireshark
i. Packet Capture Using Wireshark
ii. Starting Wireshark
iii. Viewing Captured Traffic
iv. Analysis and Statistics &Filters.

Wireshark is an open-source packet analyzer, which is used for **education, analysis, software development, communication protocol development, and network troubleshooting**.

- A *packet analyzer* is a computer application used to track, intercept and log network traffic that passes over a digital network.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a **sniffer, network protocol analyzer, and network analyzer**. It is also used by network security engineers to examine security problems.

- A **packet** is a unit of data, which is transmitted over a network between the origin and the destination.

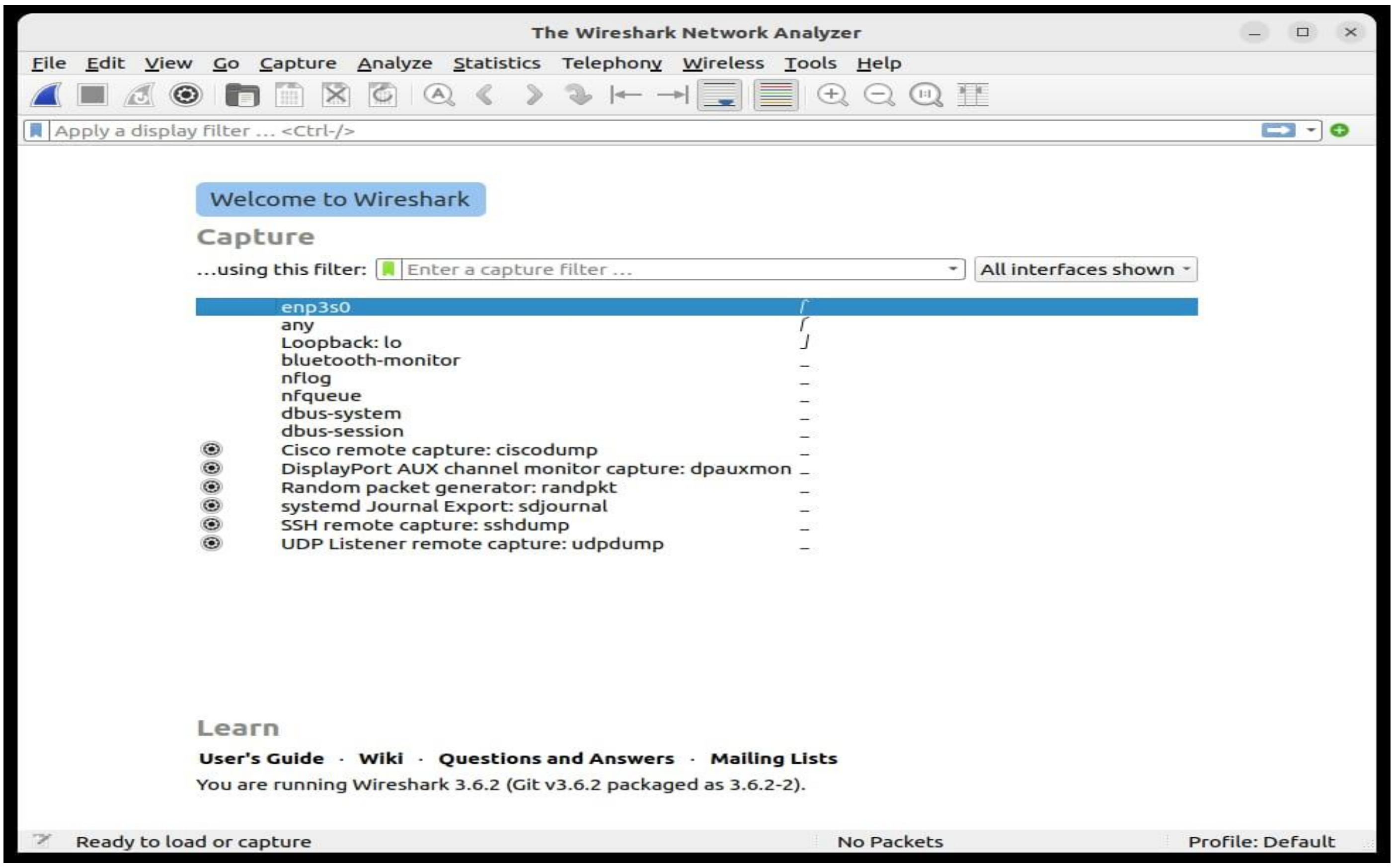
WHERE WE USE THIS WIRESHARK:

- Network administrators use it to *troubleshoot network problems*
- Network security engineers use it to *examine security problems*
- QA engineers use it to *verify network applications*
- Developers use it to *debug protocol implementations*
- People use it to *learn network protocol* internals

(i)Packet Capture using Wireshark:

After starting Wireshark, do the following:

1. Select **Capture | Interfaces**
2. Select the interface on which packets need to be captured. This will usually be the interface where the Packet/s column is constantly changing, which would indicate the presence of live traffic). If you have multiple network interface cards (i.e. LAN card and Wi-Fi adapter), you may need to check with your IT administrator to determine the right interface.
3. Click the **Start** button to start the capture.
4. Recreate the problem. The capture dialog should show the number of packets increasing. Try to avoid running any other internet applications while capturing, closing other browsers, Instant messengers etc.
5. Once the problem, which is to be analyzed, has been reproduced, click on **Stop**. It may take a few seconds for Wireshark to display the packets captured.
6. Save the packet trace in the default format. Click on the **File** menu option and select **Save As**. By default, Wireshark will save the packet trace in libpcap format. This is a filename with a.pcap extension.


The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter … <Ctrl-/>

Welcome to Wireshark

Capture

…using this filter: Enter a capture filter … All interfaces shown

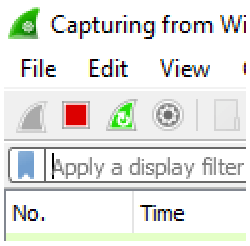
enp3s0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump

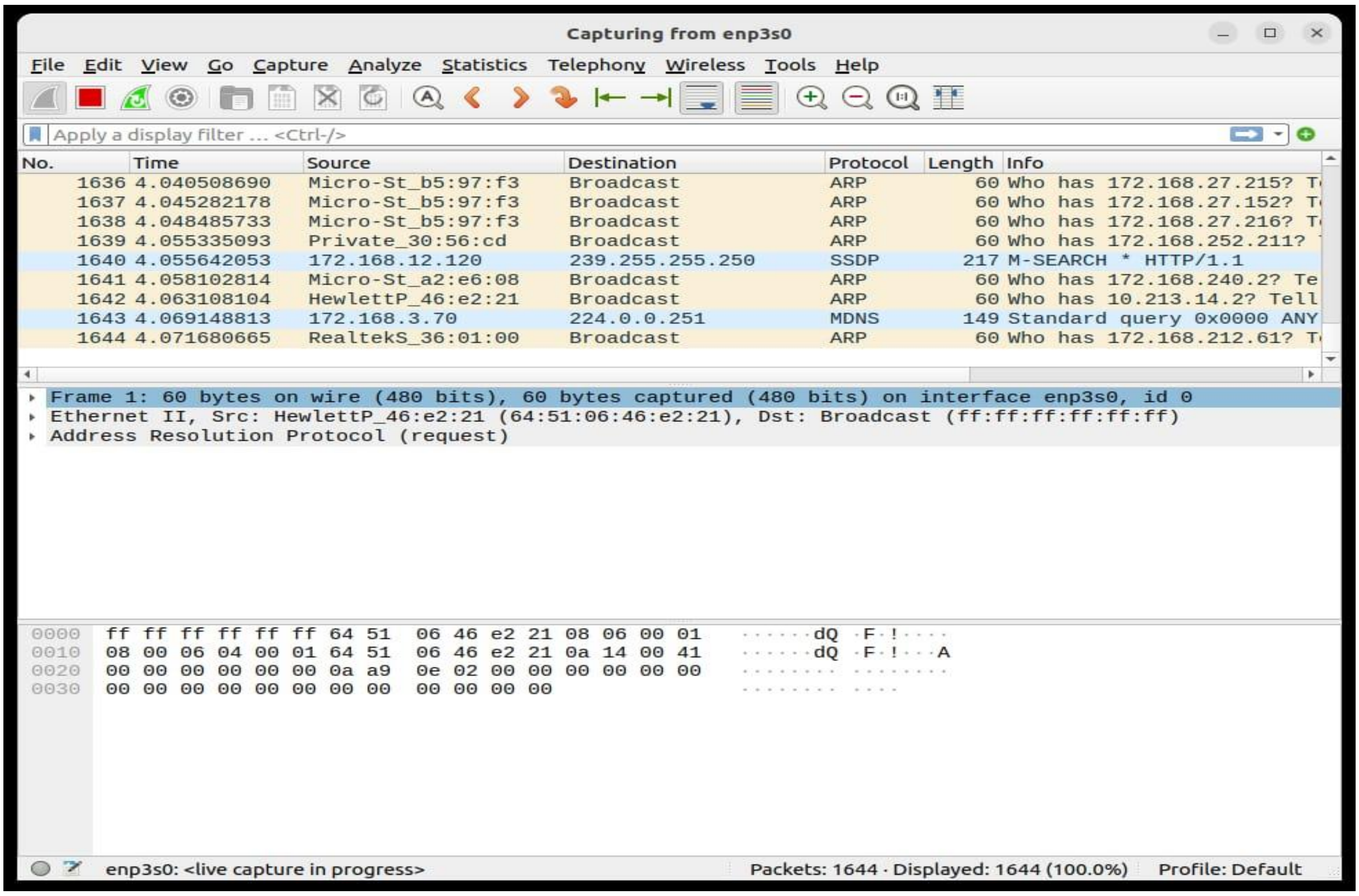
Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.6.2 (Git v3.6.2 packaged as 3.6.2-2).

Ready to load or capture No Packets Profile: Default




Capturing from enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

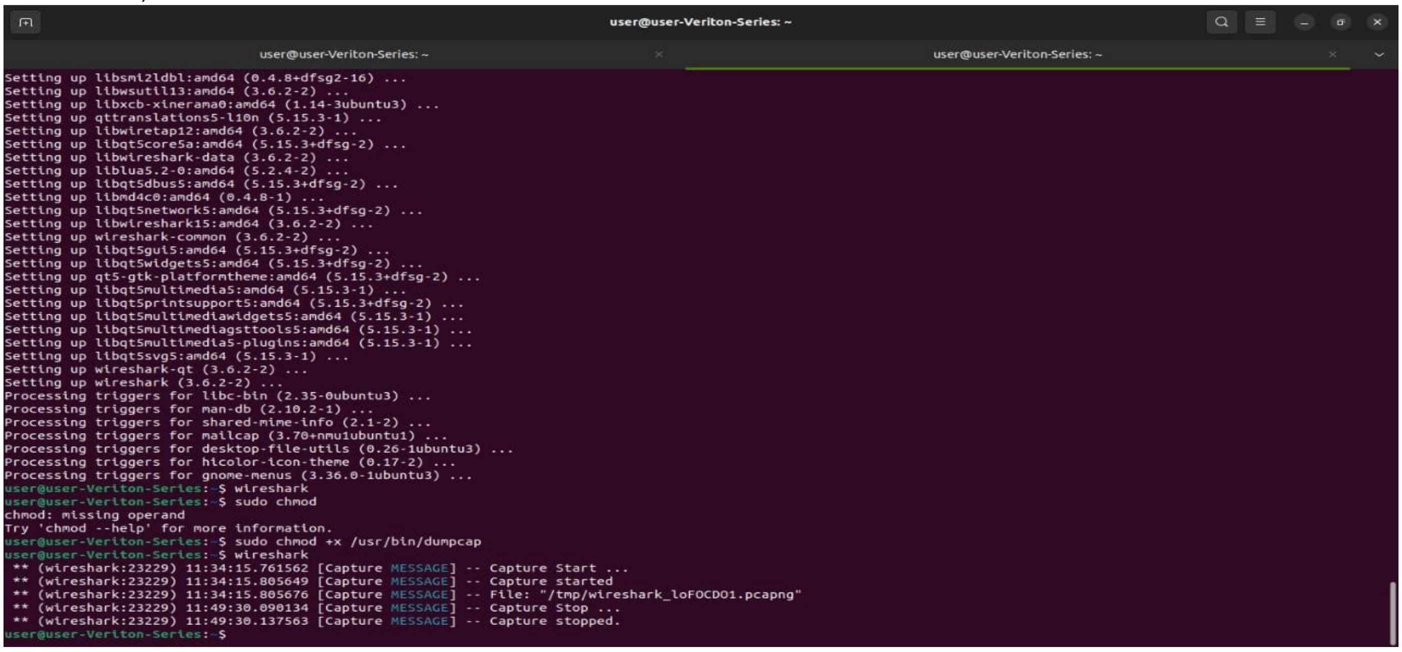
```
No.     Time          Source             Destination        Protocol Length Info
   1636 4.040508690   Micro-St_b5:97:f3  Broadcast          ARP          60 Who has 172.168.27.215? T
   1637 4.045282178   Micro-St_b5:97:f3  Broadcast          ARP          60 Who has 172.168.27.152? T
   1638 4.048485733   Micro-St_b5:97:f3  Broadcast          ARP          60 Who has 172.168.27.216? T
   1639 4.055335093   Private_30:56:cd   Broadcast          ARP          60 Who has 172.168.252.211?
   1640 4.055642053   172.168.12.120     239.255.255.250    SSDP        217 M-SEARCH * HTTP/1.1
   1641 4.058102814   Micro-St_a2:e6:08  Broadcast          ARP          60 Who has 172.168.240.2? Te
   1642 4.063108104   HewlettP_46:e2:21  Broadcast          ARP          60 Who has 10.213.14.2? Tell
   1643 4.069148813   172.168.3.70       224.0.0.251        MDNS        149 Standard query 0x0000 ANY
   1644 4.071680665   RealtekS_36:01:00  Broadcast          ARP          60 Who has 172.168.212.61? T

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp3s0, id 0
▸ Ethernet II, Src: HewlettP_46:e2:21 (64:51:06:46:e2:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 64 51  06 46 e2 21 08 06 00 01   ······dQ ·F·!····
0010  08 00 06 04 00 01 64 51  06 46 e2 21 0a 14 00 41   ······dQ ·F·!···A
0020  00 00 00 00 00 00 0a a9  0e 02 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

enp3s0: <live capture in progress>     Packets: 1644 · Displayed: 1644 (100.0%)     Profile: Default
```

(ii) To start Wireshark using the Run command box:

In windows,

1. Open the Start menu or press the Windows key + R.
2. Type Wireshark in the Run command box.
3. Press Enter

In Ubuntu,

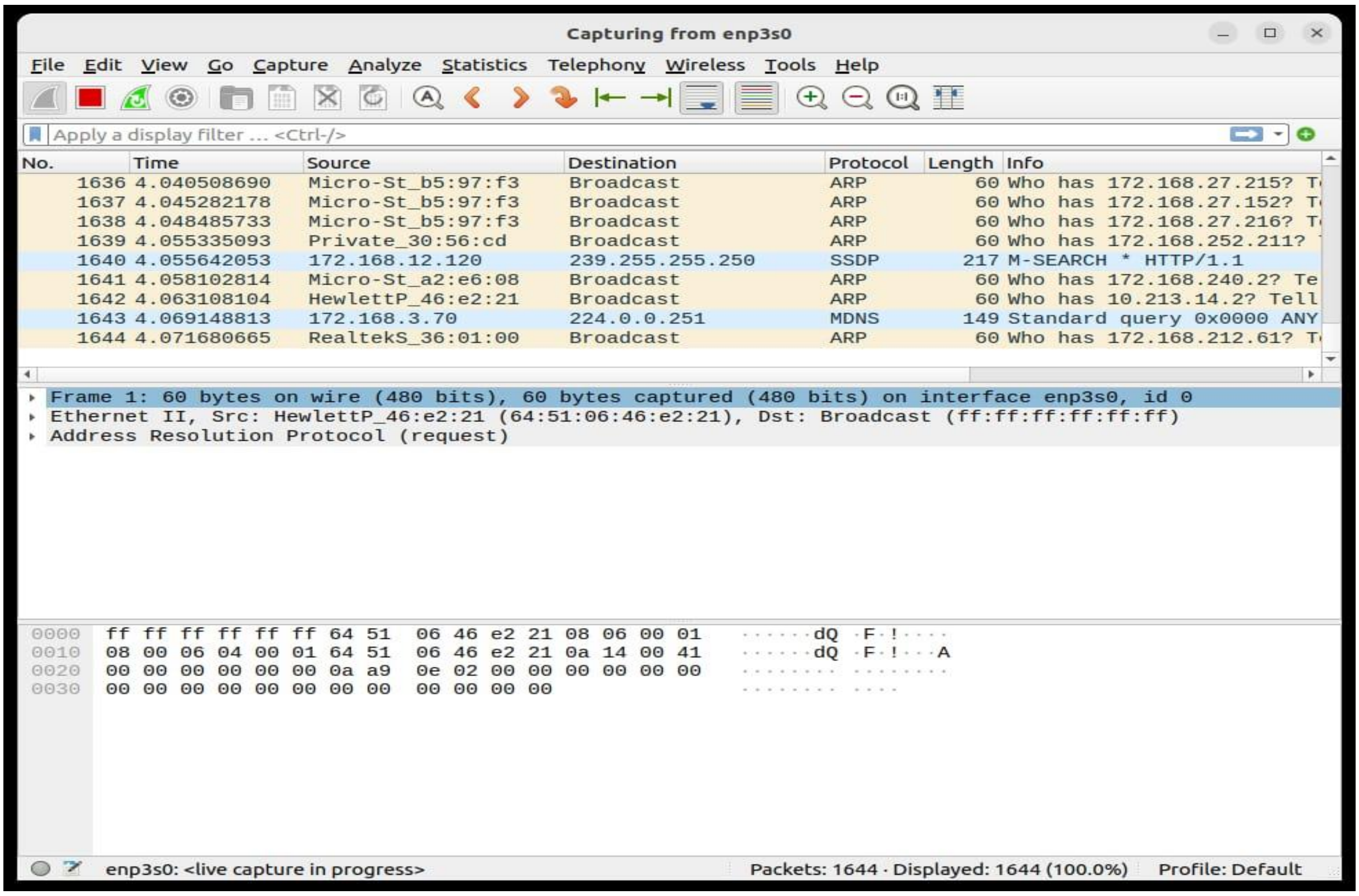

user@user-Veriton-Series: ~
user@user-Veriton-Series: ~
user@user-Veriton-Series: ~

```
Setting up libsmi2ldbl:amd64 (0.4.8+dfsg2-16) ...
Setting up libwsutil13:amd64 (3.6.2-2) ...
Setting up libxcb-xinerama0:amd64 (1.14-3ubuntu3) ...
Setting up qttranslations5-l10n (5.15.3-1) ...
Setting up libwiretap12:amd64 (3.6.2-2) ...
Setting up libqt5core5a:amd64 (5.15.3+dfsg-2) ...
Setting up libwireshark-data (3.6.2-2) ...
Setting up liblua5.2-0:amd64 (5.2.4-2) ...
Setting up libqt5dbus5:amd64 (5.15.3+dfsg-2) ...
Setting up libmd4c0:amd64 (0.4.8-1) ...
Setting up libqt5network5:amd64 (5.15.3+dfsg-2) ...
Setting up libwireshark15:amd64 (3.6.2-2) ...
Setting up wireshark-common (3.6.2-2) ...
Setting up libqt5gui5:amd64 (5.15.3+dfsg-2) ...
Setting up libqt5widgets5:amd64 (5.15.3+dfsg-2) ...
Setting up qt5-gtk-platformtheme:amd64 (5.15.3+dfsg-2) ...
Setting up libqt5multimedia5:amd64 (5.15.3-1) ...
Setting up libqt5printsupport5:amd64 (5.15.3+dfsg-2) ...
Setting up libqt5multimediawidgets5:amd64 (5.15.3-1) ...
Setting up libqt5multimediagsttools5:amd64 (5.15.3-1) ...
Setting up libqt5multimedia5-plugins:amd64 (5.15.3-1) ...
Setting up libqt5svg5:amd64 (5.15.3-1) ...
Setting up wireshark-qt (3.6.2-2) ...
Setting up wireshark (3.6.2-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
user@user-Veriton-Series:~$ wireshark
user@user-Veriton-Series:~$ sudo chmod
chmod: missing operand
Try 'chmod --help' for more information.
user@user-Veriton-Series:~$ sudo chmod +x /usr/bin/dumpcap
user@user-Veriton-Series:~$ wireshark
 ** (wireshark:23229) 11:34:15.761562 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:23229) 11:34:15.805649 [Capture MESSAGE] -- Capture started
 ** (wireshark:23229) 11:34:15.805676 [Capture MESSAGE] -- File: "/tmp/wireshark_loFOCDO1.pcapng"
 ** (wireshark:23229) 11:49:30.090134 [Capture MESSAGE] -- Capture Stop ...
 ** (wireshark:23229) 11:49:30.137563 [Capture MESSAGE] -- Capture stopped.
user@user-Veriton-Series:~$
```

(iii) Viewing Captured Traffic:

Once you have captured some packets or you have opened a previously saved capture file, you can view the packets that are displayed in the packet list pane by simply **clicking on a packet in the packet list pane**, which will bring up the selected packet in the tree view and byte view panes

.


Capturing from enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.      Time          Source               Destination        Protocol  Length Info
    1636 4.040508690   Micro-St_b5:97:f3    Broadcast          ARP           60 Who has 172.168.27.215? T
    1637 4.045282178   Micro-St_b5:97:f3    Broadcast          ARP           60 Who has 172.168.27.152? T
    1638 4.048485733   Micro-St_b5:97:f3    Broadcast          ARP           60 Who has 172.168.27.216? T
    1639 4.055335093   Private_30:56:cd     Broadcast          ARP           60 Who has 172.168.252.211?
    1640 4.055642053   172.168.12.120       239.255.255.250    SSDP         217 M-SEARCH * HTTP/1.1
    1641 4.058102814   Micro-St_a2:e6:08    Broadcast          ARP           60 Who has 172.168.240.2? Te
    1642 4.063108104   HewlettP_46:e2:21    Broadcast          ARP           60 Who has 10.213.14.2? Tell
    1643 4.069148813   172.168.3.70         224.0.0.251        MDNS         149 Standard query 0x0000 ANY
    1644 4.071680665   RealtekS_36:01:00    Broadcast          ARP           60 Who has 172.168.212.61? T

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp3s0, id 0
▸ Ethernet II, Src: HewlettP_46:e2:21 (64:51:06:46:e2:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 64 51  06 46 e2 21 08 06 00 01   ······dQ ·F·!····
0010  08 00 06 04 00 01 64 51  06 46 e2 21 0a 14 00 41   ······dQ ·F·!···A
0020  00 00 00 00 00 00 0a a9  0e 02 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

enp3s0: <live capture in progress>          Packets: 1644 · Displayed: 1644 (100.0%)     Profile: Default
```

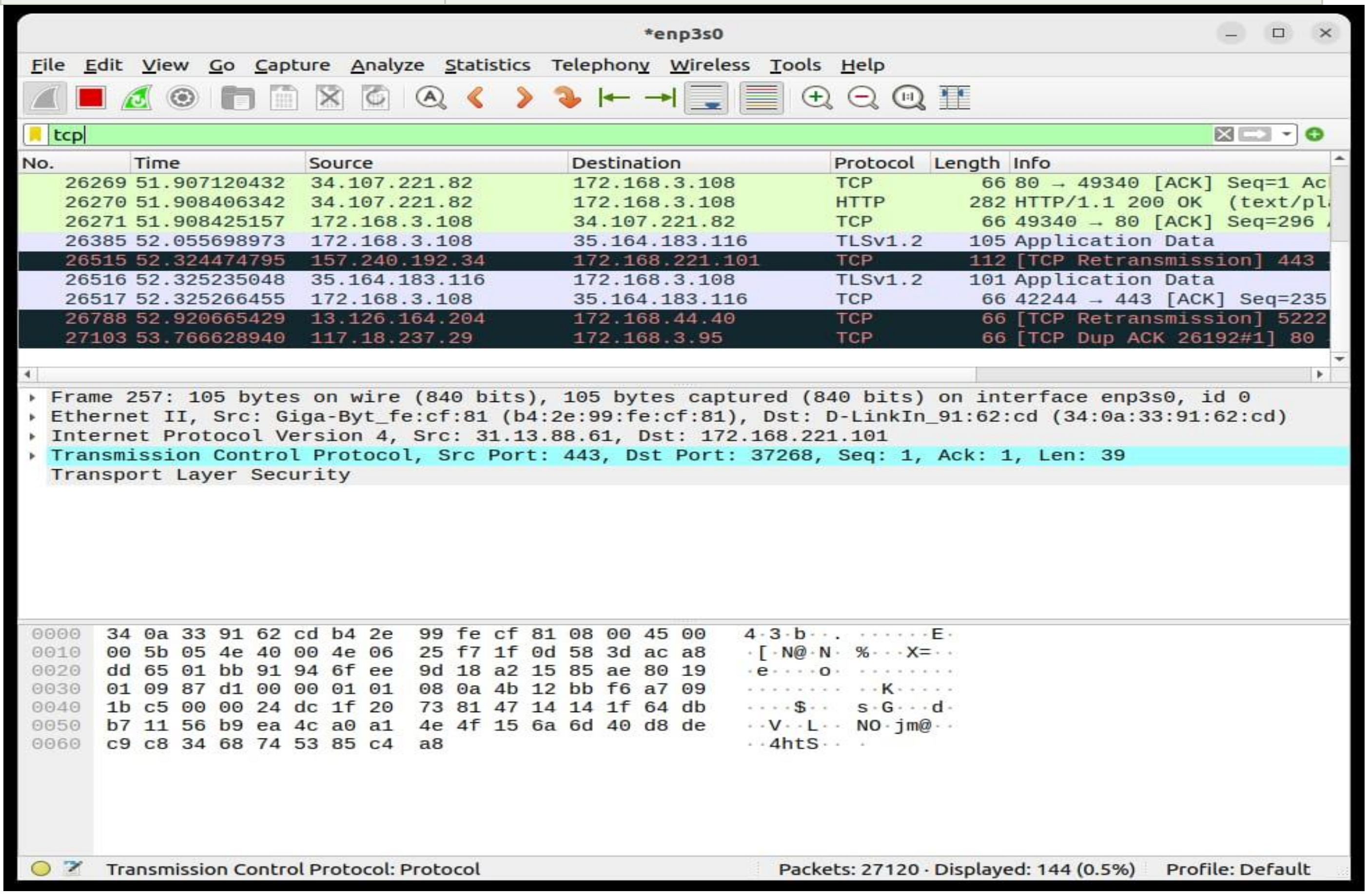
(iv) Analysis and Statistics & Filters.

Whenever we type any commands in the filter command box, it turns **green** if your command is **correct**. It turns **red** if it is **incorrect** or the Wireshark does not recognize your command.

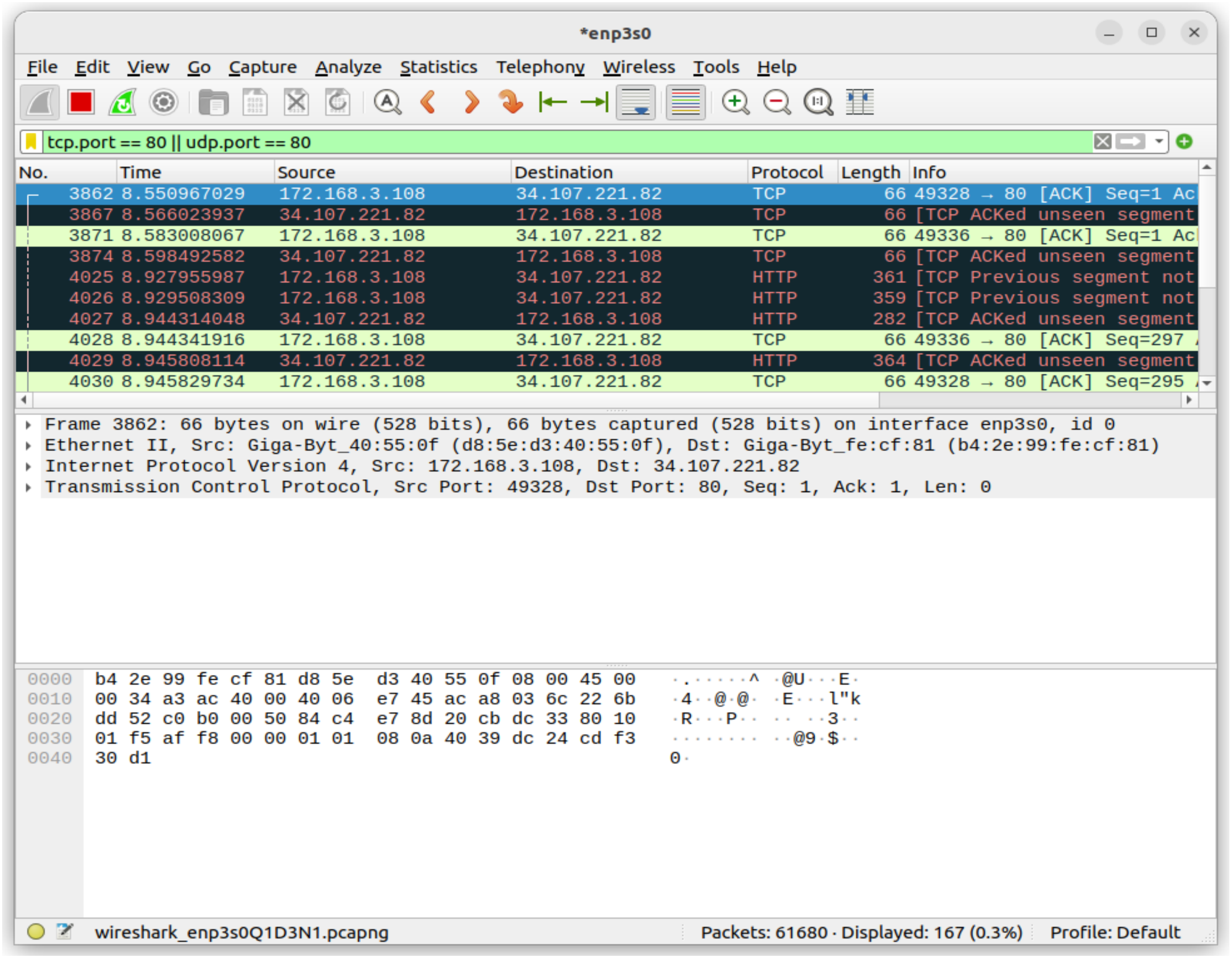
Below is the list of filters used in Wireshark:

Filters	**Description**
ip.addr Example- ip.addr==10.0.10.142 ip.src ip.dst	It is used to specify the IP address as the source or the destination. This example will filter based on this IP address as a source and a destination.If we want for a particular source or destination then, It is used for the source filter. It is used for the destination.
protocol Example- dns or http 'Dns and http' is never used.	This command filters based on the protocol. It requires the packet to be either dns protocol or http protocol and will display the traffic based on this. We would not use the command 'dns and http' because it requires the packet to be both, dns as well as http, which is impossible.
tcp.port Example: tcp.port==443	It sets filter based on the specific port number. It will filter all the packets with this port number.
4. udp.port	It is same as tcp.port. Instead, udp is used.

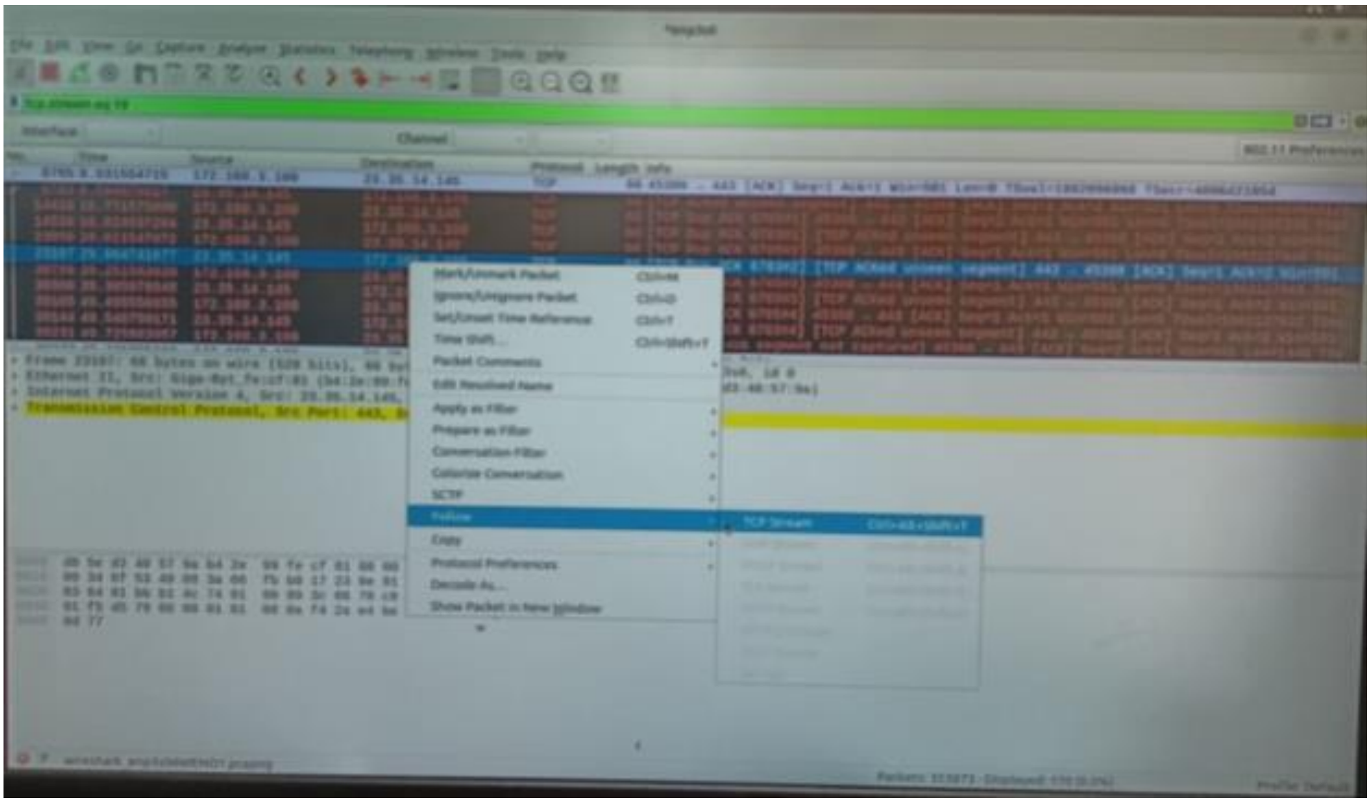
tcp.analysis.flags	Wireshark can flag TCP problems. This command will only display the issues that Wireshark identifies. Example, packet loss, tcp segment not captured, etc. are some of the problems. It quickly identifies the problem and is widely used.
6.! () For example, !(arp or dns or icmpi	It is used to filter the list of protocols or applications, in which we are not interested. It will remove arp, dns, and icmp, and only the remaining will be left or it clean the things that may not be helpful.
Select any packet. Right-click on it and select 'Follow' and then select' TCP stream.	It is used if you want to work on a single connection on a TCP conversation. Anything related to the single TCP connection will be displayed on the screen.
tcp contains the filter For example- tcp contains Facebook Orudp contains Facebook	It is used to display the packets which contain such words. In this, Facebook word in any packet in this trace file i.e., finding the devices, which are talking to Facebook.This command is useful if you are looking for a username, word, etc.
http.request For the responses or the response code, you can type http.response.code==200	It will display all the http requests in the trace file. You can see all the servers, the client is involved.
tcp.flags.syn==1 tcp.flags.reset	This will display all the packets with the sync built-in tcp header set to 1.This will show all the packets with tcp resets.

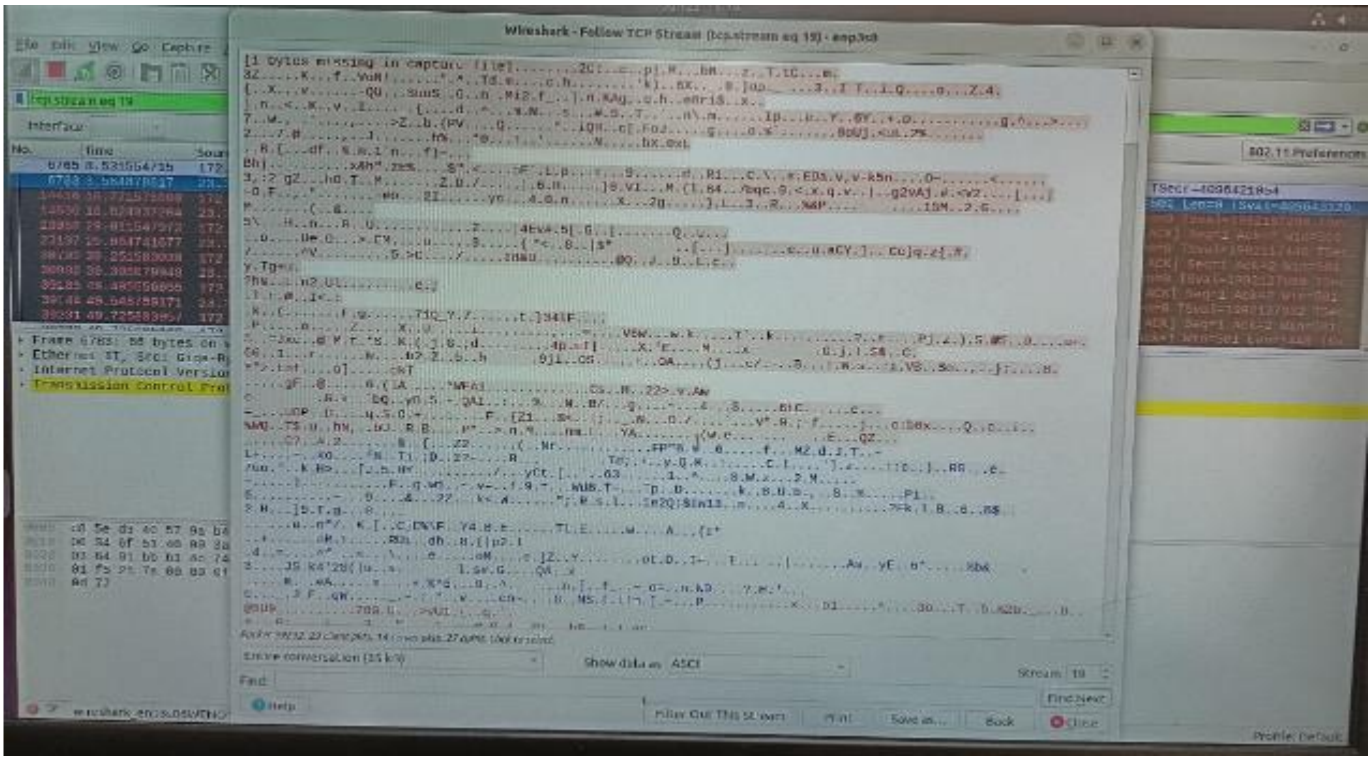

*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.    Time              Source            Destination        Protocol  Length Info
 26269 51.907120432   34.107.221.82     172.168.3.108      TCP          66 80 → 49340 [ACK] Seq=1 Ac
 26270 51.908406342   34.107.221.82     172.168.3.108      HTTP        282 HTTP/1.1 200 OK  (text/pl
 26271 51.908425157   172.168.3.108     34.107.221.82      TCP          66 49340 → 80 [ACK] Seq=296
 26385 52.055698973   172.168.3.108     35.164.183.116     TLSv1.2     105 Application Data
 26515 52.324474795   157.240.192.34    172.168.221.101    TCP         112 [TCP Retransmission] 443
 26516 52.325235048   35.164.183.116    172.168.3.108      TLSv1.2     101 Application Data
 26517 52.325266455   172.168.3.108     35.164.183.116     TCP          66 42244 → 443 [ACK] Seq=235
 26788 52.920665429   13.126.164.204    172.168.44.40      TCP          66 [TCP Retransmission] 5222
 27103 53.766628940   117.18.237.29     172.168.3.95       TCP          66 [TCP Dup ACK 26192#1] 80

▸ Frame 257: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface enp3s0, id 0
▸ Ethernet II, Src: Giga-Byt_fe:cf:81 (b4:2e:99:fe:cf:81), Dst: D-LinkIn_91:62:cd (34:0a:33:91:62:cd)
▸ Internet Protocol Version 4, Src: 31.13.88.61, Dst: 172.168.221.101
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 37268, Seq: 1, Ack: 1, Len: 39
  Transport Layer Security

0000  34 0a 33 91 62 cd b4 2e  99 fe cf 81 08 00 45 00   4·3·b··. ······E·
0010  00 5b 05 4e 40 00 4e 06  25 f7 1f 0d 58 3d ac a8   ·[·N@·N· %···X=··
0020  dd 65 01 bb 91 94 6f ee  9d 18 a2 15 85 ae 80 19   ·e····o· ········
0030  01 09 87 d1 00 00 01 01  08 0a 4b 12 bb f6 a7 09   ········ ··K·····
0040  1b c5 00 00 24 dc 1f 20  73 81 47 14 14 1f 64 db   ····$··  s·G···d·
0050  b7 11 56 b9 ea 4c a0 a1  4e 4f 15 6a 6d 40 d8 de   ··V··L·· NO·jm@··
0060  c9 c8 34 68 74 53 85 c4  a8                        ··4htS·· ·

Transmission Control Protocol: Protocol      Packets: 27120 · Displayed: 144 (0.5%)      Profile: Default
```


*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.port == 80 || udp.port == 80

No.  Time         Source          Destination     Protocol Length Info
3862 8.550967029  172.168.3.108   34.107.221.82   TCP      66 49328 → 80 [ACK] Seq=1 Ac
3867 8.566023937  34.107.221.82   172.168.3.108   TCP      66 [TCP ACKed unseen segment
3871 8.583008067  172.168.3.108   34.107.221.82   TCP      66 49336 → 80 [ACK] Seq=1 Ac
3874 8.598492582  34.107.221.82   172.168.3.108   TCP      66 [TCP ACKed unseen segment
4025 8.927955987  172.168.3.108   34.107.221.82   HTTP     361 [TCP Previous segment not
4026 8.929508309  172.168.3.108   34.107.221.82   HTTP     359 [TCP Previous segment not
4027 8.944314048  34.107.221.82   172.168.3.108   HTTP     282 [TCP ACKed unseen segment
4028 8.944341916  172.168.3.108   34.107.221.82   TCP      66 49336 → 80 [ACK] Seq=297 
4029 8.945808114  34.107.221.82   172.168.3.108   HTTP     364 [TCP ACKed unseen segment
4030 8.945829734  172.168.3.108   34.107.221.82   TCP      66 49328 → 80 [ACK] Seq=295 

Frame 3862: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp3s0, id 0
Ethernet II, Src: Giga-Byt_40:55:0f (d8:5e:d3:40:55:0f), Dst: Giga-Byt_fe:cf:81 (b4:2e:99:fe:cf:81)
Internet Protocol Version 4, Src: 172.168.3.108, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 49328, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

0000  b4 2e 99 fe cf 81 d8 5e  d3 40 55 0f 08 00 45 00   ·.·····^ ·@U···E·
0010  00 34 a3 ac 40 00 40 06  e7 45 ac a8 03 6c 22 6b   ·4··@·@· ·E···l"k
0020  dd 52 c0 b0 00 50 84 c4  e7 8d 20 cb dc 33 80 10   ·R···P·· ·· ··3··
0030  01 f5 af f8 00 00 01 01  08 0a 40 39 dc 24 cd f3   ········ ··@9·$··
0040  30 d1                                              0·

wireshark_enp3s0Q1D3N1.pcapng     Packets: 61680 · Displayed: 167 (0.3%)     Profile: Default
```





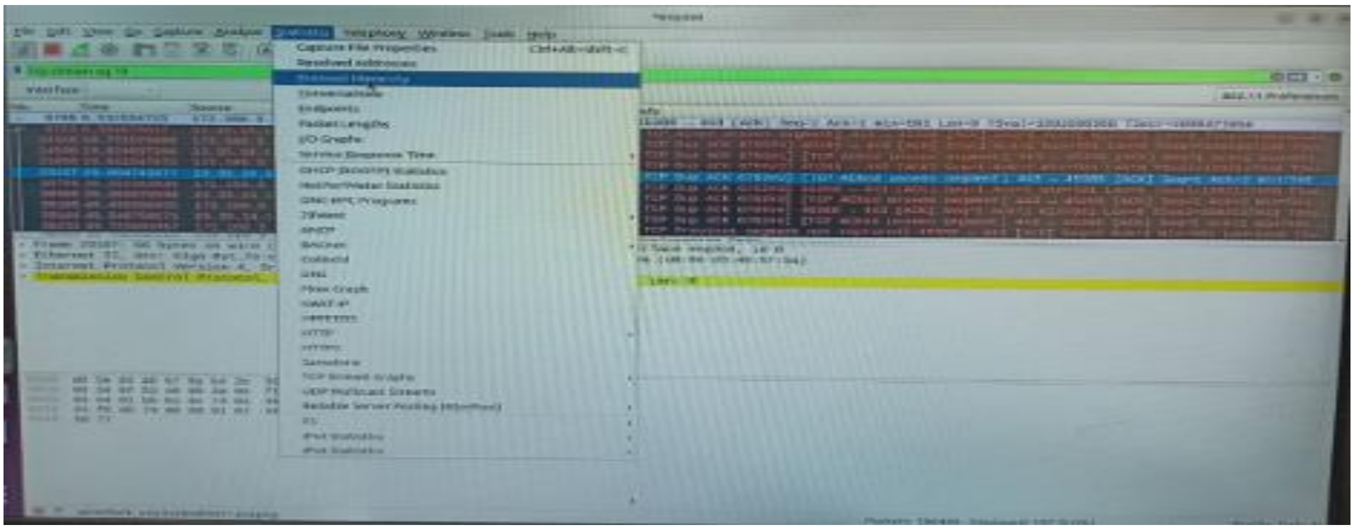
Wireshark Statistics

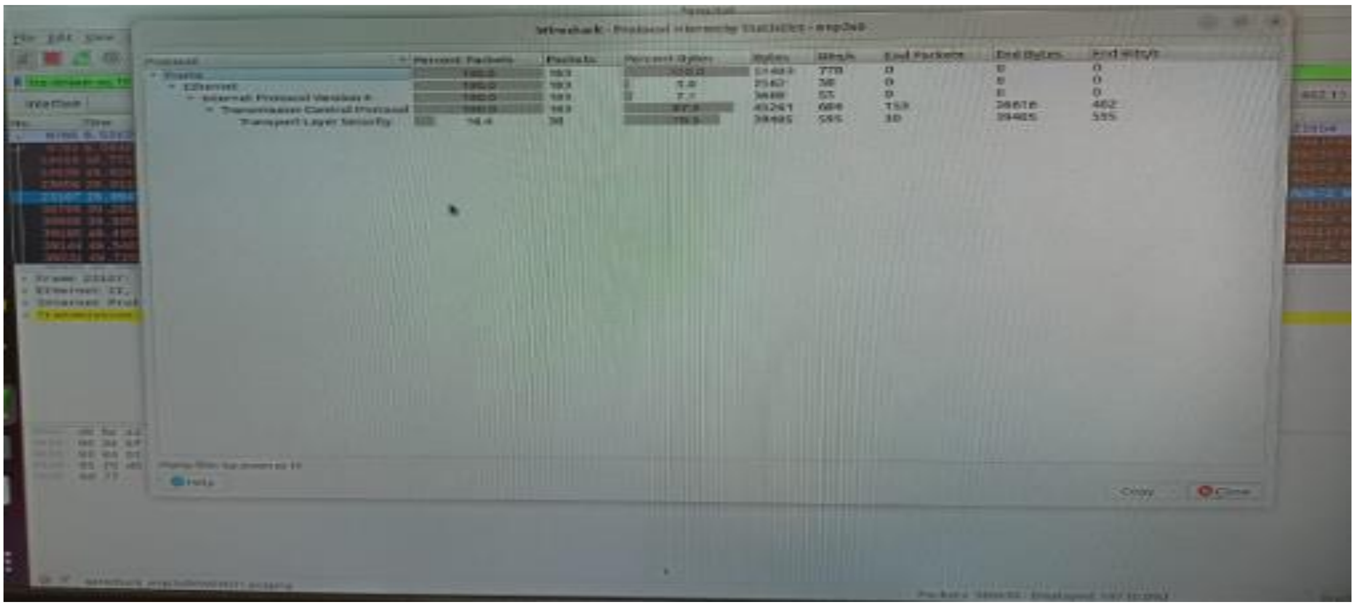
The Wireshark provides a wide domain of statistics. They are listed below:
Below is the list of statistics of Wireshark along with the description:

Capture file properties	It includes file, time, capture, interfaces (current interface in use), and Statistics (measurements).
Resolved addresses	This option includes all the types of the Top IP addresses and DNS that were resolved in your packet capture. It gives the idea of the different accessed resources during the packet capture process.
Protocol hierarchy	It is named as the tree of all the protocols listed in the capture process.
Conversations	Each row of the list gives the statistical value of a particular conversation.
Endpoints	It is defined as a logical endpoint of the separate protocol traffic of the specified protocol layer. For example IP address will send and receive all types of the packet to the particular IP addresses.
Packet lengths	It simply displays the characteristics of different packets lengths determined in the network.
I/O Graphs	It is the term used to display the graph of the captured packets. You can also apply filters during this process. The process is explained below in detail.
Service Response Time	It is the type of information which is available for many protocols. It is defined as the time it takes between the request and the response time. The protocol for which this service is available are: AFP (Apple Filing Protocol) CAMEL DCE-RPC DIAMETER

	FC (Fiber Channel) GTP (GPRS Tunneling Protocol) H.225 RAS LDAP (Lightweight Directory Access Protocol) MEGACO MGCP (Media Gateway Control Protocol) NCP (NetWare Core Protocol) ONC-RPC RADIUS SCSI SMB (Server Message Block Protocol) SMB2 (Server Message Block Protocol version 2)
DHCP (BOOTP) Statistics	It is implemented as the option of BOOTP. DHCP is client/server protocol, dynamically used to assign IP addresses to a DHCP client. If DHCP does not work, then some computer system uses APIPA (Automatic Private IP Address) to assign the IP addresses.
ONC-RPC Programs	It stands for Open Network Computing- Remote Procedure Call. It can use TCP and UDP as its transport protocol. ONC-RPC cannot be applied directly to filter in a capture process, but you can use TCP or UDP to filter on that one.
29West	It is defined as ULLM technology. It stands for Ultra-Low Latency Messaging.
ANCP	It stands for **Access Node Control Protocol**. It is an L2CP (Layer 2 Control Protocol) and a TCP based one. It has its adjacency layer which decides the messages exchange by the ANCP endpoints with the use of 'Capabilities.'
BACnet	It was designed specially to meet the communication needs of control systems and building automation. It is used for applications such as fire detecting systems, light control, etc. It provides the structure to exchange information despite the particular building service it performs.
Collectd	It is used to monitor the traffic on the specific TCP port.
DNS	It stands for Domain Name Server, which gives a detailed analysis of the DNS traffic. It provides the list of the codes returned in DNS. You can also view the errors through the traffic.
Flow-graph	It is a method to check connections between the client and the server. It is an efficient way to verify the connections between two endpoints. It also assists us with troubleshooting capabilities.
HART-IP	It gives the detail for the response, request, publishes, and error packets. It stands for Highway Addressable Remote Transducer over IP stats.
HPFEEDS	It determines the 'payload size per channel and Opcodes.'
HTTP	It has four options: ○ Packet counter (request types and response codes) ○ Requests (based on URL and the host) ○ Load distribution (based on server address and host)

	○ Request sequences (sequences the HTTP's capture request as a tree)
HTTP2	It is the HTTP version 2.
Sametime	It is used to analyze the slow network traffic when the server and client have the sametime.
TCP Stream Graphs	It is explained below in detail:
UDP Multicast Streams	Through this command, stream parameters and burst parameters can be set. It includes OSPF, IGMP, and video streams.
F5	It includes the virtual server distribution and the tmm distribution. It specifies the tcpdump commands.
IPv4 Statistics IPv6 Statistics	These options determine all addresses, destination and ports, IP protocol types, and the source and destination address.





11. How to run N map scan

- **Nmap** is a free and open-source network discovery and security auditing utility that is widely used in the Linux users community as it is simple to use yet very powerful.

(or)

- **Nmap** is a Linux command-line tool for network exploration and security auditing. This tool is generally used by hackers and cybersecurity enthusiasts and even by network and system administrators.
- It is used for the following purposes:
  - Real time information of a network
  - Detailed information of all the IPs activated on your network
  - Number of ports open in a network
  - Provide the list of live hosts
  - Port, OS and Host scanning

Install NmapOn Ubuntu

First, you should get the "nmap" package installed in your system.

sudo apt-get install nmap

Using the Nmap security scanner

- Then you may run the command "nmap" on a terminal, accompanied by the target's IP or website address and the various available parameters. To get an overview of all the parameters that nmap can be used with, use the "nmap –help" command.

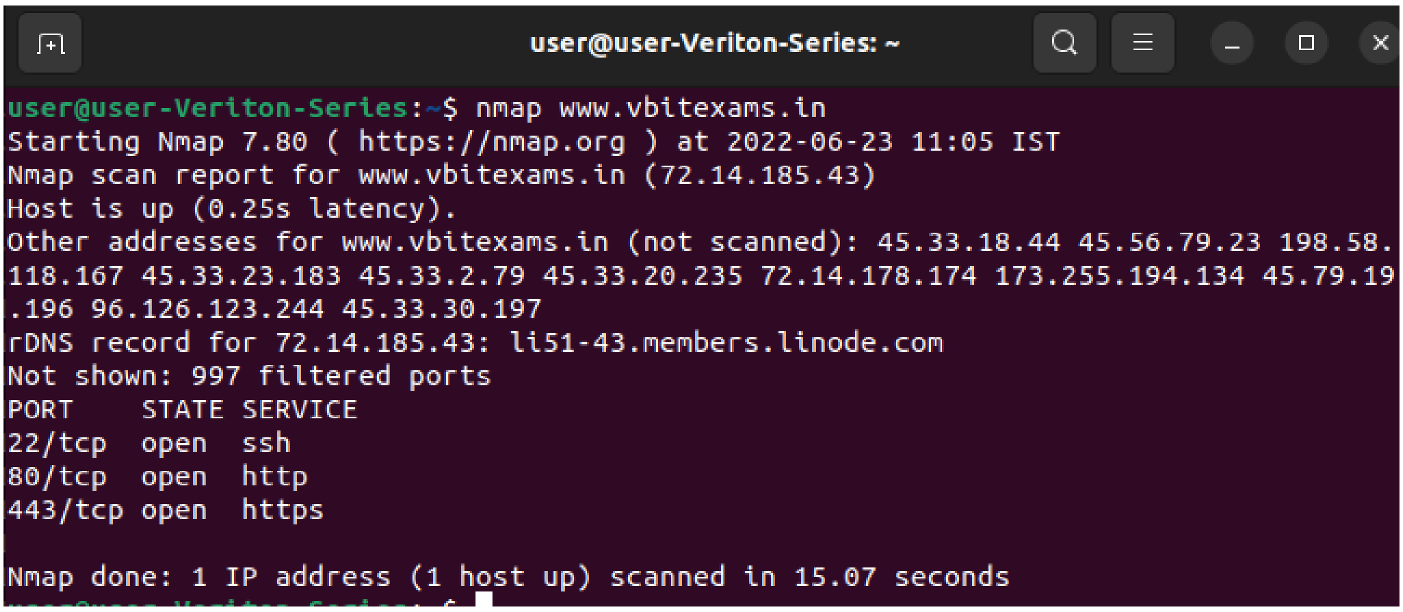
- To scan a System with Hostname and IP address,will give all open ports, services,and MAC addresses on the system.

  First, Scan using Hostname

  Example: *nmap*[*www.geeksforgeeks.org*](http://www.geeksforgeeks.org)

Now let usscan using IP Address

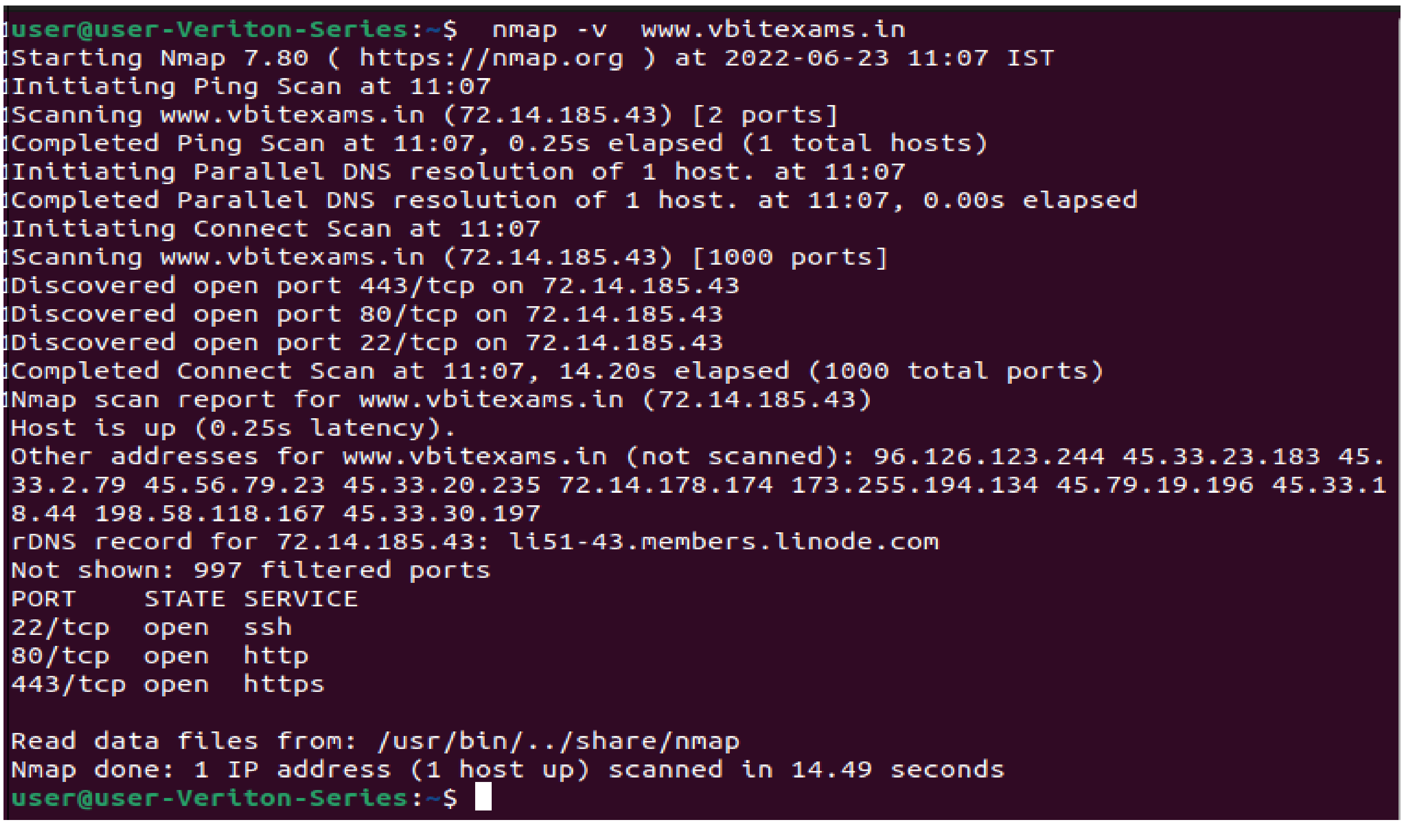
Example: *nmap 172.217.27.174*



```
user@user-Veriton-Series:~$ nmap www.vbitexams.in
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:05 IST
Nmap scan report for www.vbitexams.in (72.14.185.43)
Host is up (0.25s latency).
Other addresses for www.vbitexams.in (not scanned): 45.33.18.44 45.56.79.23 198.58.
118.167 45.33.23.183 45.33.2.79 45.33.20.235 72.14.178.174 173.255.194.134 45.79.19
.196 96.126.123.244 45.33.30.197
rDNS record for 72.14.185.43: li51-43.members.linode.com
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 15.07 seconds
```

- To scan using –v option,give more detailed information about the remote machines.

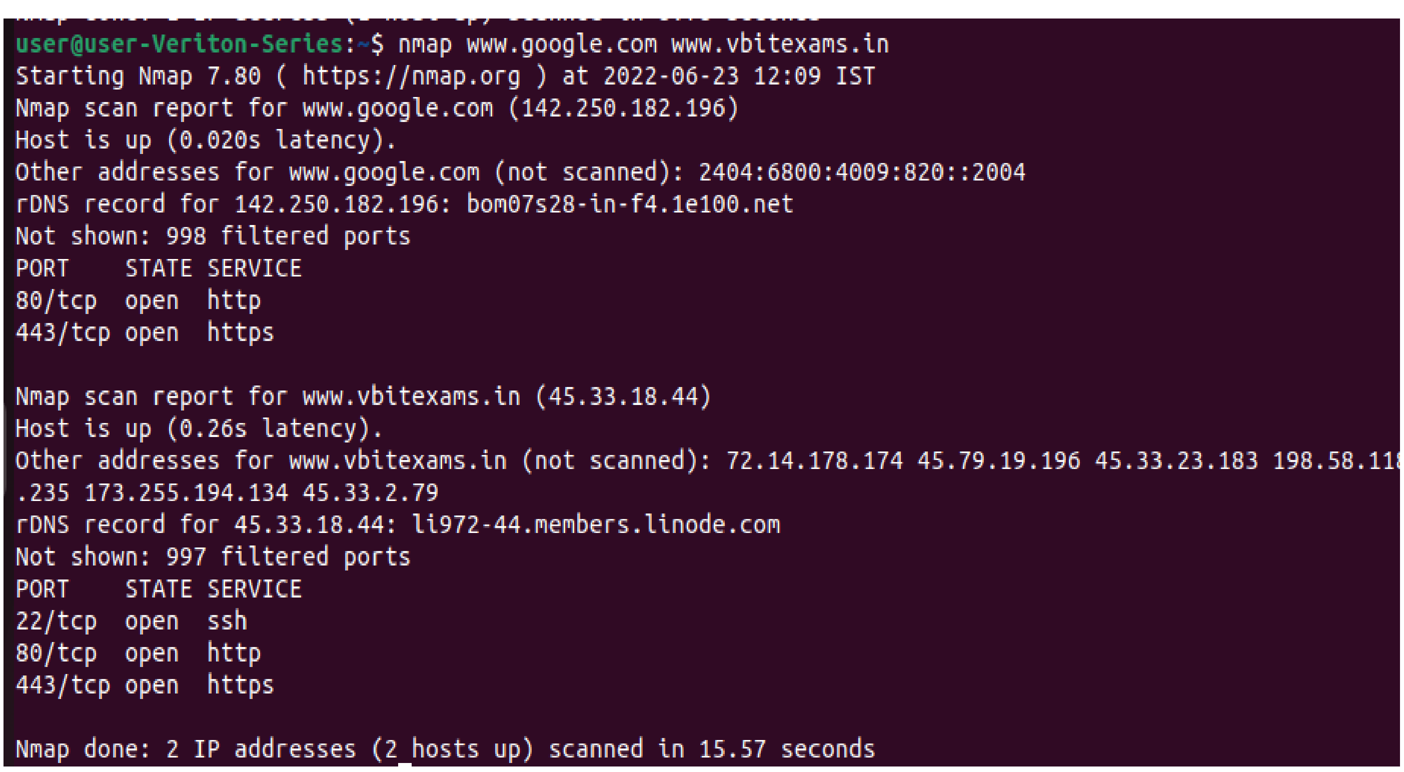
  nmap –v [IP address] or [website address]



```
user@user-Veriton-Series:~$  nmap -v  www.vbitexams.in
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:07 IST
Initiating Ping Scan at 11:07
Scanning www.vbitexams.in (72.14.185.43) [2 ports]
Completed Ping Scan at 11:07, 0.25s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:07
Completed Parallel DNS resolution of 1 host. at 11:07, 0.00s elapsed
Initiating Connect Scan at 11:07
Scanning www.vbitexams.in (72.14.185.43) [1000 ports]
Discovered open port 443/tcp on 72.14.185.43
Discovered open port 80/tcp on 72.14.185.43
Discovered open port 22/tcp on 72.14.185.43
Completed Connect Scan at 11:07, 14.20s elapsed (1000 total ports)
Nmap scan report for www.vbitexams.in (72.14.185.43)
Host is up (0.25s latency).
Other addresses for www.vbitexams.in (not scanned): 96.126.123.244 45.33.23.183 45.
33.2.79 45.56.79.23 45.33.20.235 72.14.178.174 173.255.194.134 45.79.19.196 45.33.1
8.44 198.58.118.167 45.33.30.197
rDNS record for 72.14.185.43: li51-43.members.linode.com
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.49 seconds
user@user-Veriton-Series:~$
```

- To scan multiple hosts

  nmap 103.76.228.244 157.240.198.35 172.217.27.174

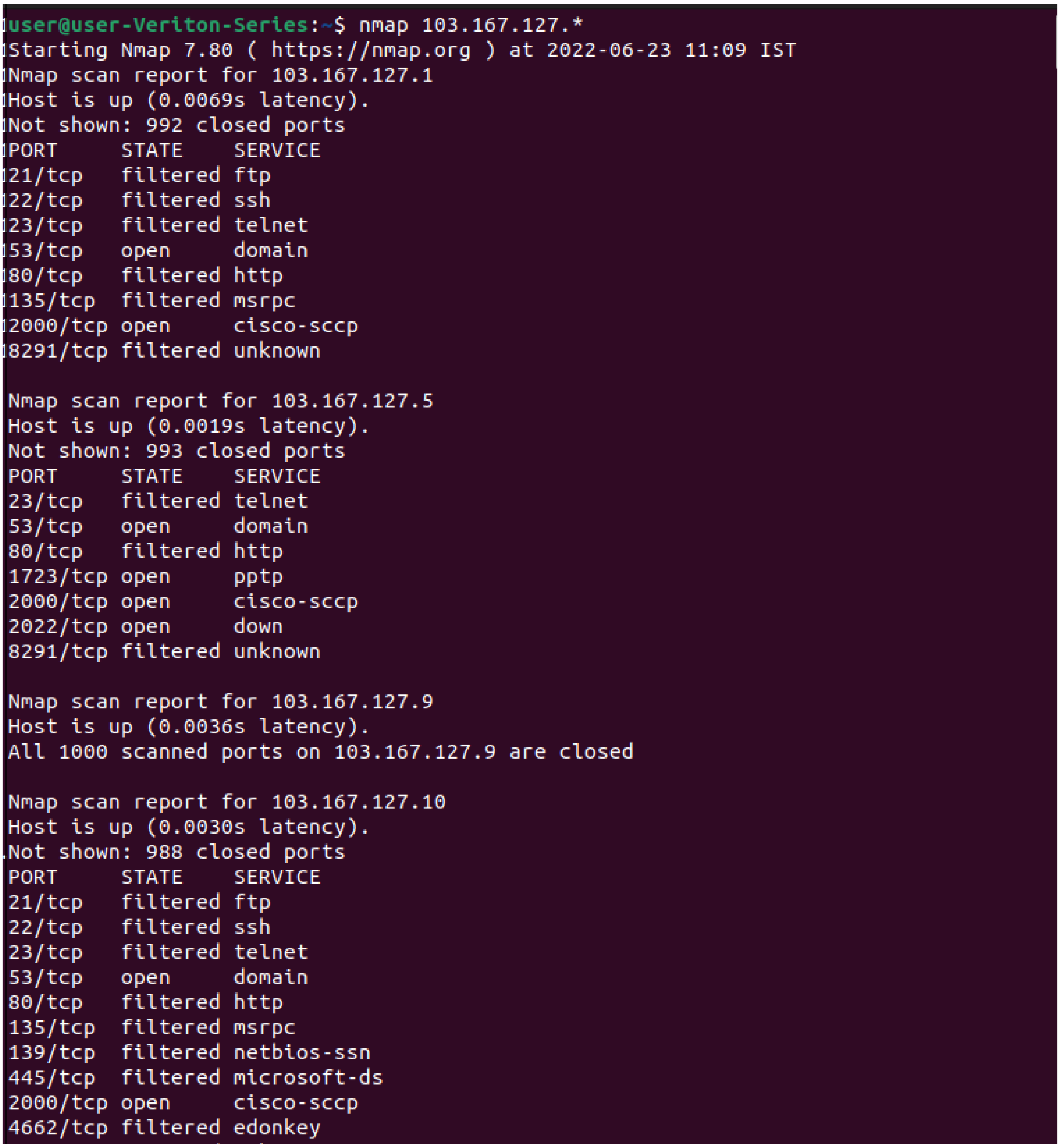


```
user@user-Veriton-Series:~$ nmap www.google.com www.vbitexams.in
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 12:09 IST
Nmap scan report for www.google.com (142.250.182.196)
Host is up (0.020s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:820::2004
rDNS record for 142.250.182.196: bom07s28-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for www.vbitexams.in (45.33.18.44)
Host is up (0.26s latency).
Other addresses for www.vbitexams.in (not scanned): 72.14.178.174 45.79.19.196 45.33.23.183 198.58.11
.235 173.255.194.134 45.33.2.79
rDNS record for 45.33.18.44: li972-44.members.linode.com
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 2 IP addresses (2 hosts up) scanned in 15.57 seconds
```

- To scan whole subnet

  *nmap 103.76.228.**

  We can scan a whole subnet or IP range with nmap by providing “*” with it. It will scan a whole subnet and give the information about those hosts, which are Up in the Network.

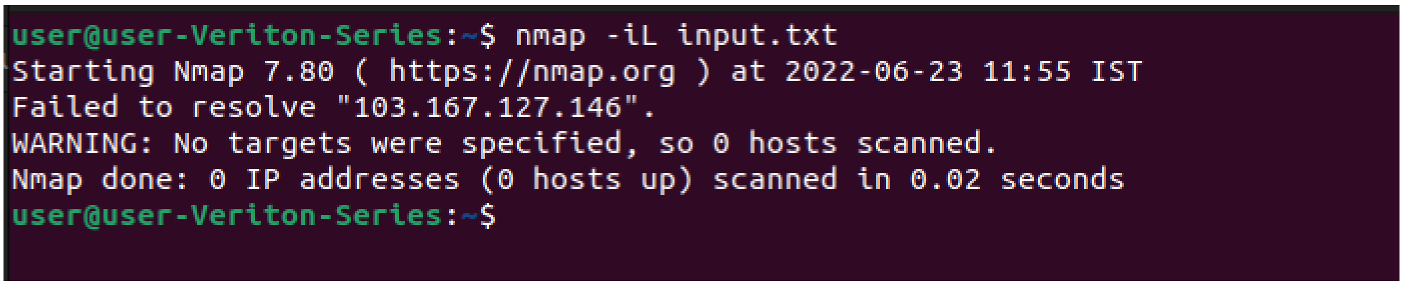


```
user@user-Veriton-Series:~$ nmap 103.167.127.*
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:09 IST
Nmap scan report for 103.167.127.1
Host is up (0.0069s latency).
Not shown: 992 closed ports
PORT     STATE    SERVICE
21/tcp   filtered ftp
22/tcp   filtered ssh
23/tcp   filtered telnet
53/tcp   open     domain
80/tcp   filtered http
135/tcp  filtered msrpc
2000/tcp open     cisco-sccp
8291/tcp filtered unknown

Nmap scan report for 103.167.127.5
Host is up (0.0019s latency).
Not shown: 993 closed ports
PORT     STATE    SERVICE
23/tcp   filtered telnet
53/tcp   open     domain
80/tcp   filtered http
1723/tcp open     pptp
2000/tcp open     cisco-sccp
2022/tcp open     down
8291/tcp filtered unknown

Nmap scan report for 103.167.127.9
Host is up (0.0036s latency).
All 1000 scanned ports on 103.167.127.9 are closed

Nmap scan report for 103.167.127.10
Host is up (0.0030s latency).
Not shown: 988 closed ports
PORT     STATE    SERVICE
21/tcp   filtered ftp
22/tcp   filtered ssh
23/tcp   filtered telnet
53/tcp   open     domain
80/tcp   filtered http
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
2000/tcp open     cisco-sccp
4662/tcp filtered edonkey
```

- To scan from a file

  nmap -iL input.txt

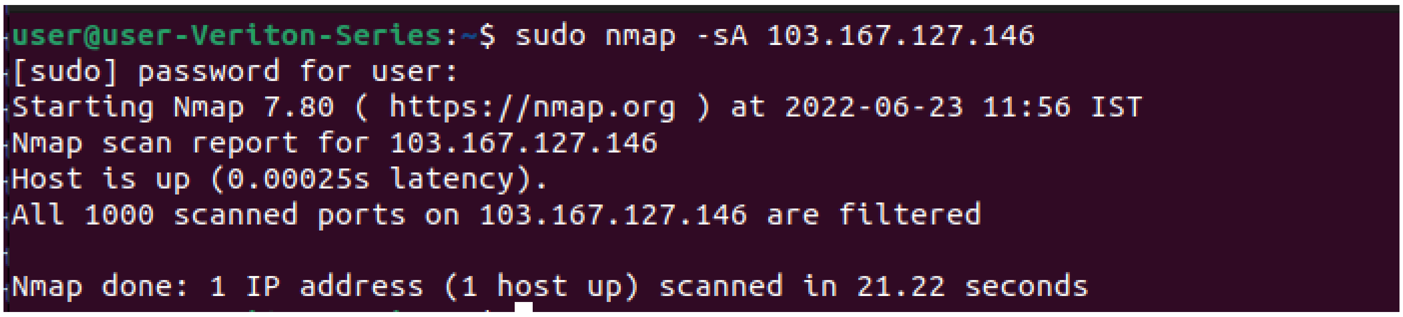
If we have a long list of addresses that we need to scan, we can directly import a file through the command line. It will produce a scan for the given IP addresses.


```
user@user-Veriton-Series:~$ nmap -iL input.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:55 IST
Failed to resolve "103.167.127.146".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.02 seconds
user@user-Veriton-Series:~$
```

- To scan to detect firewall settings.

  sudonmap -sA 103.76.228.244

Detecting firewall settings can be useful during penetration testing and vulnerability scans. To detect it we use "-sA" option. This will provide you with information about firewall being active on the host. It uses an ACK scan to receive the information.


```
user@user-Veriton-Series:~$ sudo nmap -sA 103.167.127.146
[sudo] password for user:
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:56 IST
Nmap scan report for 103.167.127.146
Host is up (0.00025s latency).
All 1000 scanned ports on 103.167.127.146 are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.22 seconds
```

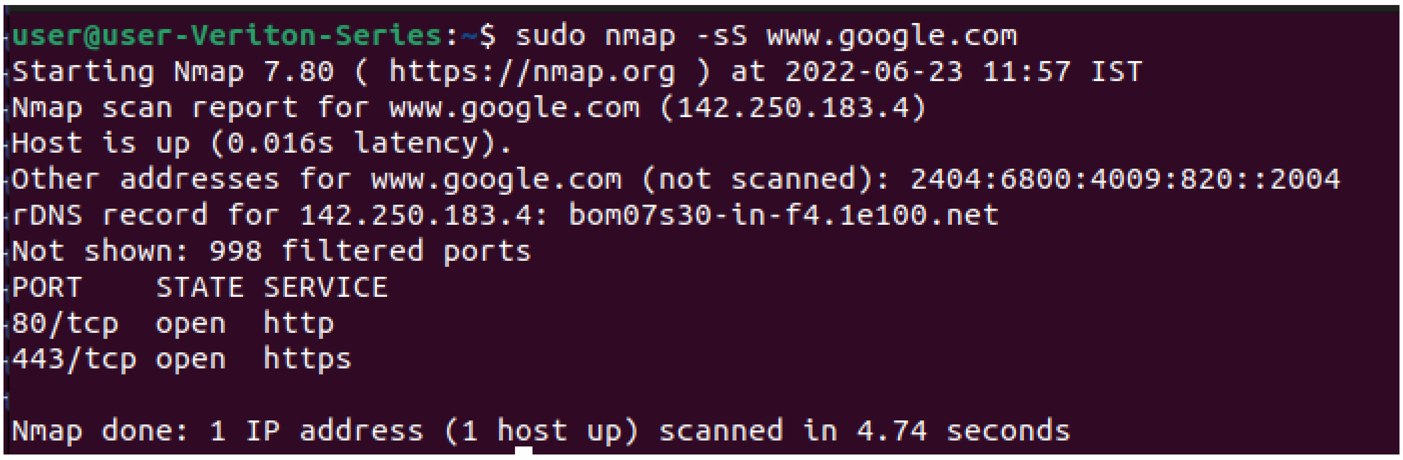
- To identify Hostnames

  sudonmap -sL 103.76.228.244

We use "sL" option to find hostnames for the given host by completing a DNS query for each one. In addition to this "-n" command can be used to skip DNS resolution, while the "-R" command can be used to always resolve DNS.

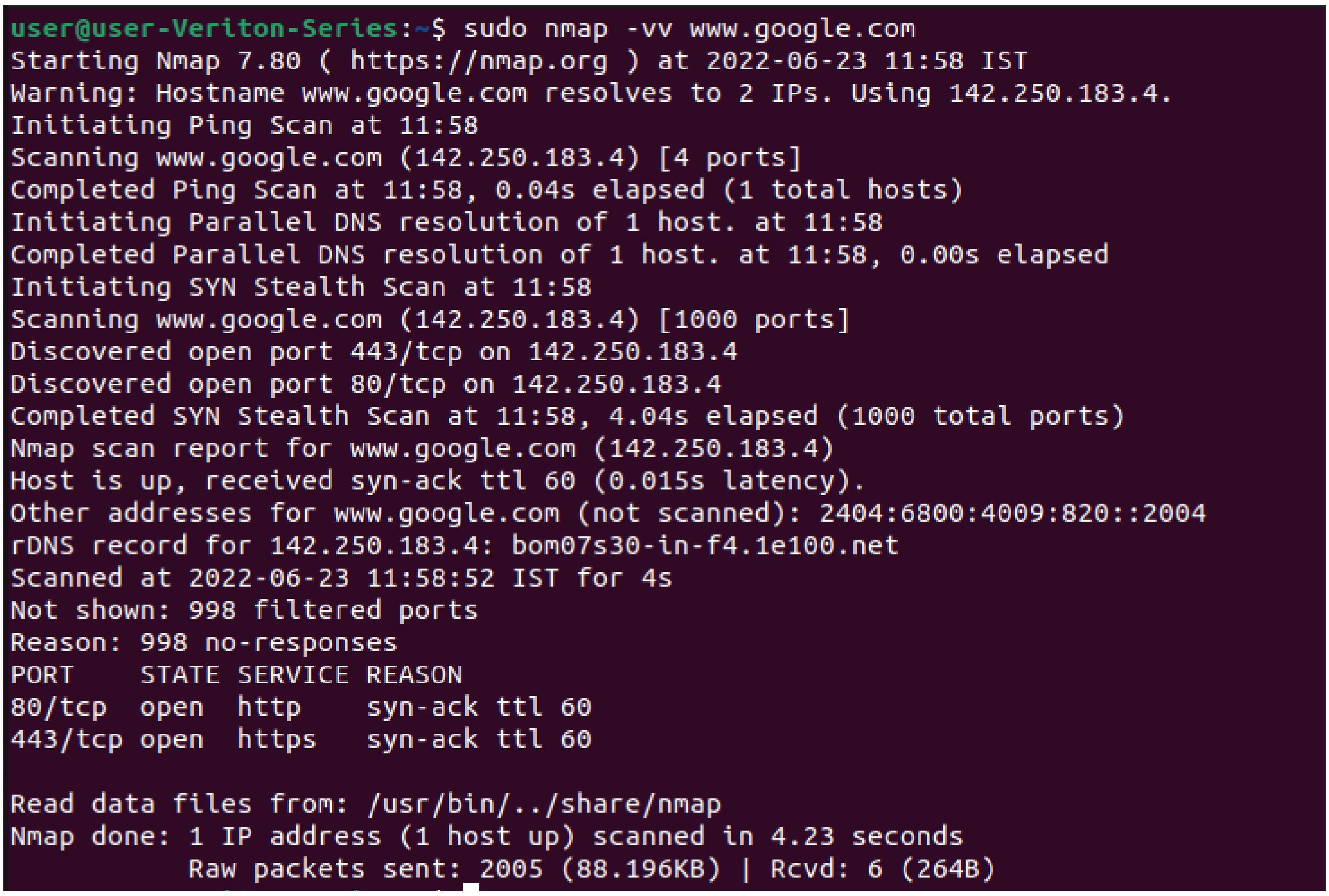
sudonmap –sS [IP address] or [website address]

- This is a basic command that can be used to quickly check if a website is up without creating any "loggable" interaction with the target server


```
user@user-Veriton-Series:~$ sudo nmap -sS www.google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:57 IST
Nmap scan report for www.google.com (142.250.183.4)
Host is up (0.016s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:820::2004
rDNS record for 142.250.183.4: bom07s30-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.74 seconds
```

sudonmap –vv [IP address] or [website address]

- This is a ping scan command that is helpful when wanting to check the open ports of a target.



```
user@user-Veriton-Series:~$ sudo nmap -vv www.google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 11:58 IST
Warning: Hostname www.google.com resolves to 2 IPs. Using 142.250.183.4.
Initiating Ping Scan at 11:58
Scanning www.google.com (142.250.183.4) [4 ports]
Completed Ping Scan at 11:58, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:58
Completed Parallel DNS resolution of 1 host. at 11:58, 0.00s elapsed
Initiating SYN Stealth Scan at 11:58
Scanning www.google.com (142.250.183.4) [1000 ports]
Discovered open port 443/tcp on 142.250.183.4
Discovered open port 80/tcp on 142.250.183.4
Completed SYN Stealth Scan at 11:58, 4.04s elapsed (1000 total ports)
Nmap scan report for www.google.com (142.250.183.4)
Host is up, received syn-ack ttl 60 (0.015s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:820::2004
rDNS record for 142.250.183.4: bom07s30-in-f4.1e100.net
Scanned at 2022-06-23 11:58:52 IST for 4s
Not shown: 998 filtered ports
Reason: 998 no-responses
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 60
443/tcp open  https   syn-ack ttl 60

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.23 seconds
           Raw packets sent: 2005 (88.196KB) | Rcvd: 6 (264B)
```

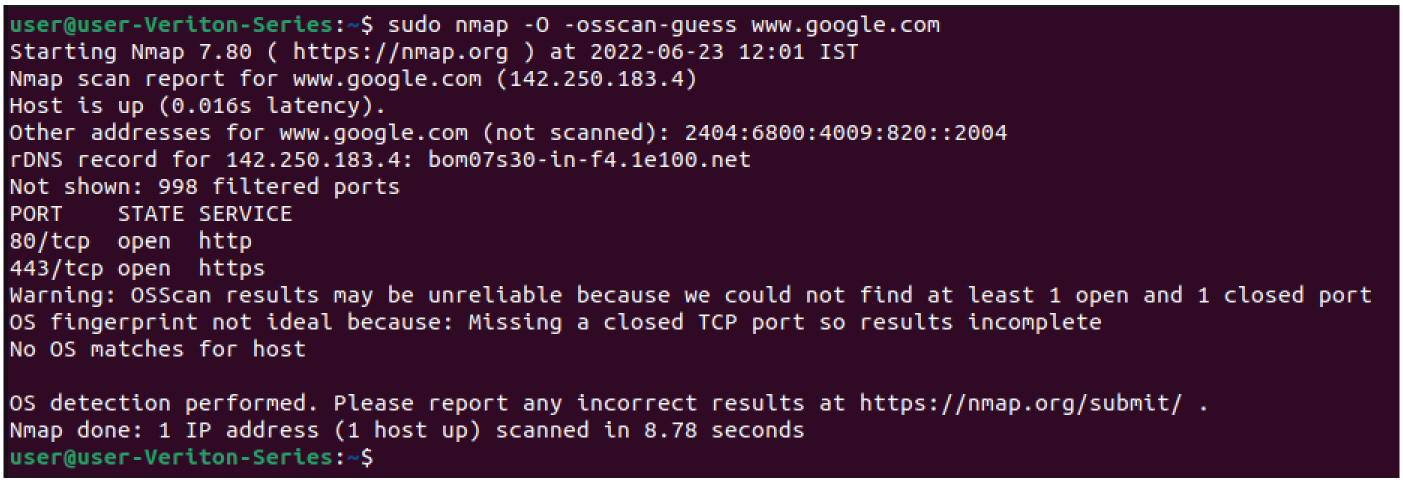
sudonmap –sV [IP address] or [website address]

- ➢ This command allows the user to check the services run on the target. Note that the “cloudflare-ngnix” information was added next to the services of the open ports.

12. Operating System Detection using Nmap

sudonmap –O –osscan-guess [IP address] or [website address]

- ➢ This command instructs nmap to try to guess what operating system is run on the target system. If all ports are filtered, this “guessing” command will be the best option, but the results cannot be considered reliable. The percentage numbers are giving the idea of uncertainty.



```
user@user-Veriton-Series:~$ sudo nmap -O -osscan-guess www.google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-23 12:01 IST
Nmap scan report for www.google.com (142.250.183.4)
Host is up (0.016s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:820::2004
rDNS record for 142.250.183.4: bom07s30-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.78 seconds
user@user-Veriton-Series:~$
```

13. Do the following using NS2Simulator

i. NS2Simulator-Introduction

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to substantial contributions from the players in the field. Among these are the University of California and Cornell University who developed the REAL network simulator,1 the foundation which NS is based on. Since 1995 the Defense Advanced Research Projects Agency (DARPA) supported development of NS through the Virtual Inter Network Testbed (VINT) project . Currently the National Science Foundation (NSF) has joined the ride in development. Last but not the least, the group of Researchers and developers in the community are constantly working to keep NS2 strong and versatile.
BASIC ARCHITECTURE:

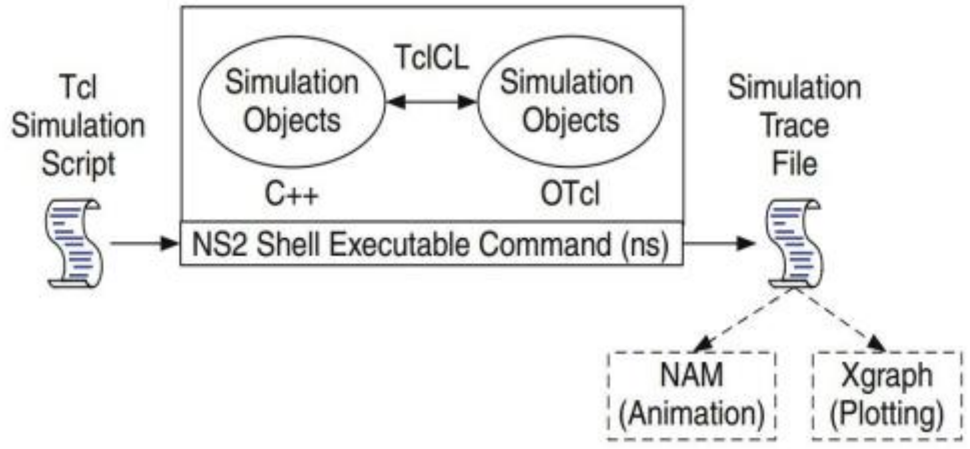


Figure 2.1 shows the basic architecture of NS2. NS2 provides users with an executable command ns which takes on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g.,_o10) in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTclobjects. It may defines its own procedures and variables to facilitate the interaction. Note that the member procedures and variables in the OTcl domain are called instance procedures (instprocs) and instance variables (instvars), respectively. Before proceeding further, the readers are encouraged to learn C++ and OTcl languages. We refer the readers to [14] for the detail of C++, while a brief tutorial of Tcl and OTcl tutorial are given in Appendices A.1 and A.2, respectively. NS2 provides a large number of built-in C++ objects. It is advisable to use these C++ objects to set up a simulation using a Tcl simulation script. However, advance users may find these objects insufficient. They need to develop their own C++ objects, and use a OTcl configuration interface to put together these objects. After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and

interactively, tools such as NAM (Network AniMator) and XGraph are used. To analyze a particular behaviour of the network, users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.

ii. Simulate to Find the Number of Packets Dropped

In Network simulator 2 trace file generally;

Column 1 represents event type (s: send, r: receive, d: drop, f: forward)

Columns 2 and 3 provide time of the event.

Columns 4 to 7 provide next-hop information.

Columns 8 to 21 are called node property tags.

Column 11 gives the reason for a packet drop and the various reasons possible for a packet drop are as follows.

END drop due to end of simulation

COL drop due to collision at MAC layer

DUP drop due to duplicate packet

ERR drop due to MAC packet error

RET drop due to exceeding retry count

STA drop due to MAC invalid state

BSY drop due to MAC busy

NRTE drop due to no route available

LOOP drop due to routing loop

TTL drop due to TTL=0

TOUT drop due to packet expired

IFQ drop due to no buffer space in IFQ

ARP dropped ARP

OUT dropped by base stations

You may analyze your dropped packets by grouping them with the reasons.

```
BEGIN {
```

```
sendLine = 0;

recvLine = 0;

fowardLine = 0;

dropLine = 0;

}



$0 ~/^s.* AGT/ {

sendLine ++ ;

}



$0 ~/^r.* AGT/ {

recvLine ++ ;

}



$0 ~/^f.* RTR/ {

fowardLine ++ ;

}


$0 ~/^D.* cbr/ {

dropLine ++ ;

}

END {
printf "Packet Sent: %d\n,sendLine"
printf "Packet Received: %d\n,recvLine"
printf "Packet Drop: %d\n,dropLine"


printf "PacketDeliveryRatio:%.4f \n",(recvLine/sendLine);
}
```

iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
iv. Simulate to Find the Number of Packets Dropped due to Congestion
v. Simulate to Compare Data Rate & Throughput.
vi. Simulate to Plot Congestion for Different Source/Destination
vii. Simulate to Determine the Performance with respect to Transmission of Packet