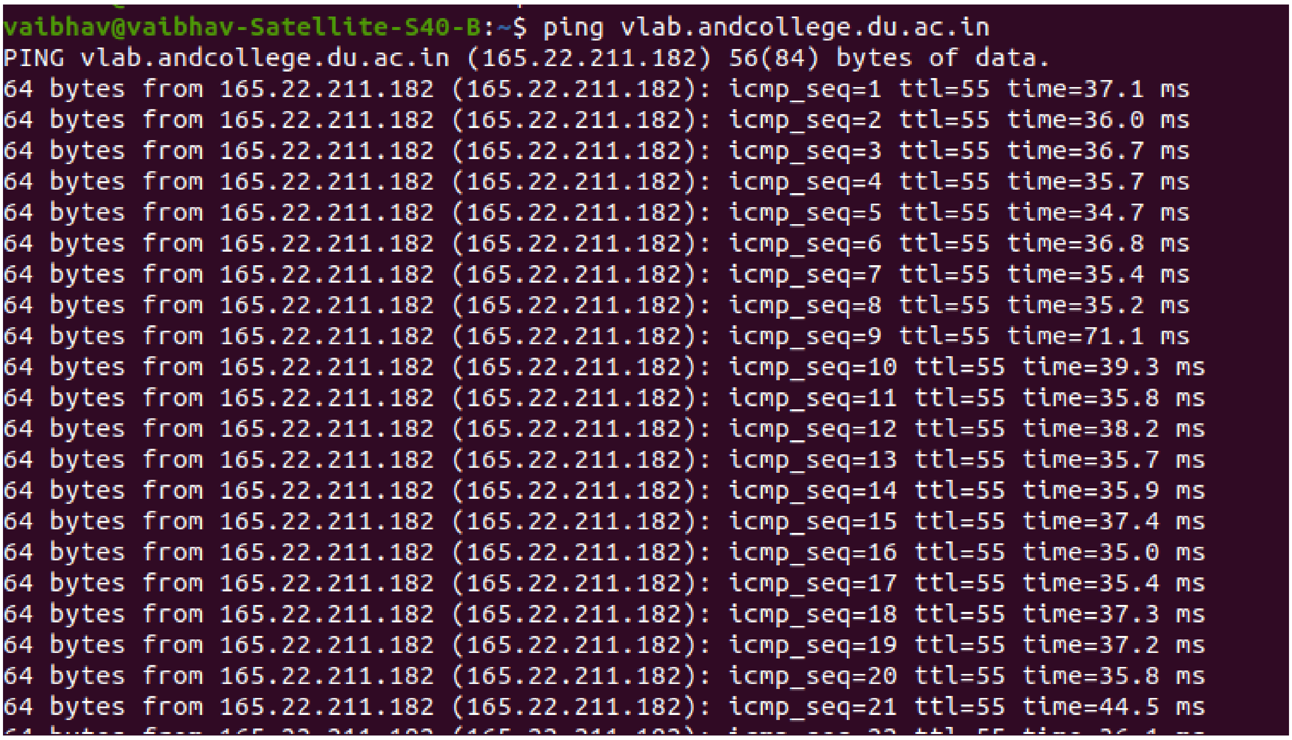
Name – Vaibhav Thapliyal
College Roll. No. - AC-859
Exam Roll No. - 20001567045
Subject- GE-IV (Information Security and Cyber Laws)

Practical file

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.

Use of ping:

Ping tool is used to test whether a particular host is reachable across an IP address or not.



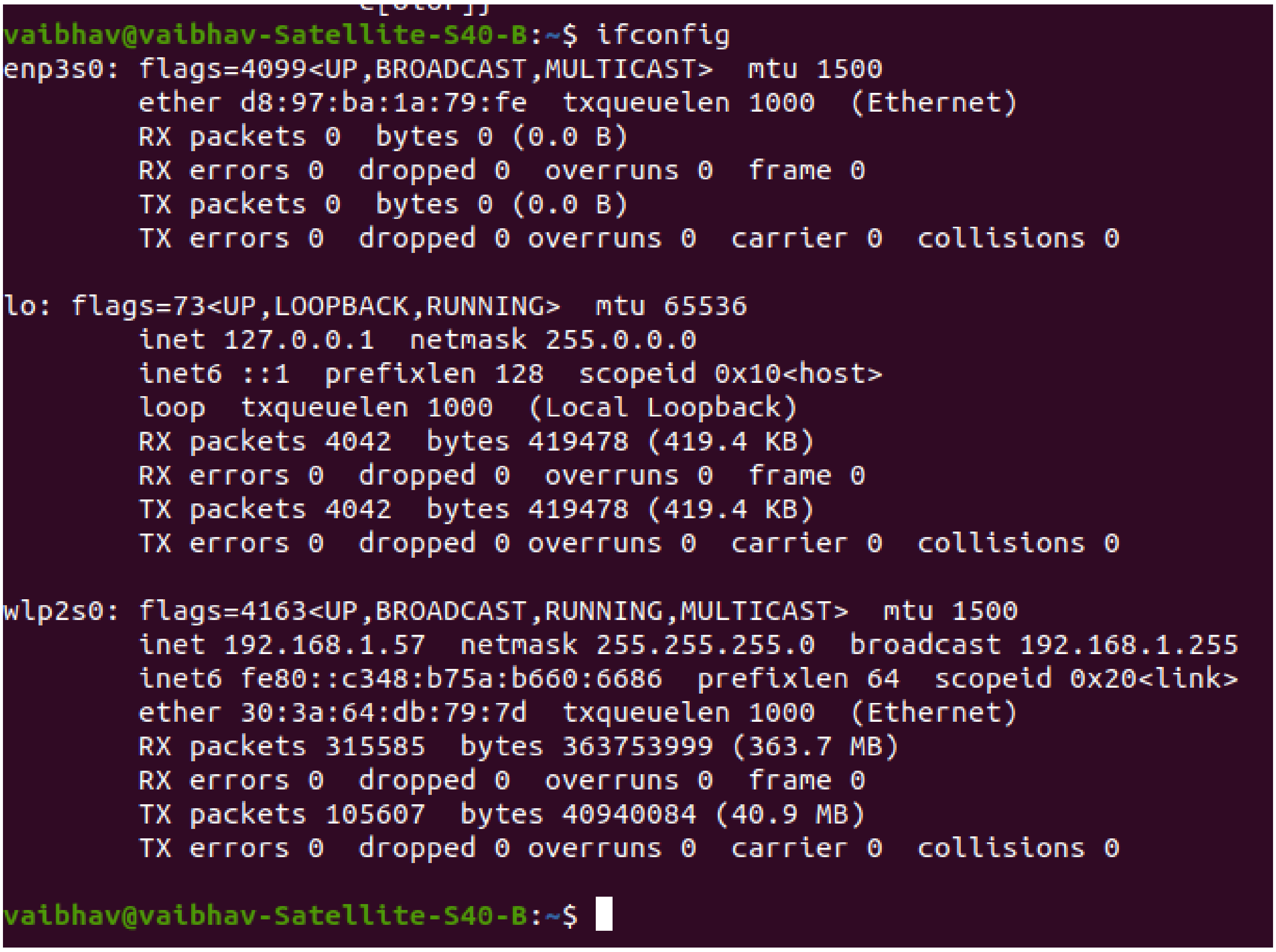
```
vaibhav@vaibhav-Satellite-S40-B:~$ ping vlab.andcollege.du.ac.in
PING vlab.andcollege.du.ac.in (165.22.211.182) 56(84) bytes of data.
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=1 ttl=55 time=37.1 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=2 ttl=55 time=36.0 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=3 ttl=55 time=36.7 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=4 ttl=55 time=35.7 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=5 ttl=55 time=34.7 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=6 ttl=55 time=36.8 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=7 ttl=55 time=35.4 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=8 ttl=55 time=35.2 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=9 ttl=55 time=71.1 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=10 ttl=55 time=39.3 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=11 ttl=55 time=35.8 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=12 ttl=55 time=38.2 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=13 ttl=55 time=35.7 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=14 ttl=55 time=35.9 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=15 ttl=55 time=37.4 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=16 ttl=55 time=35.0 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=17 ttl=55 time=35.4 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=18 ttl=55 time=37.3 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=19 ttl=55 time=37.2 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=20 ttl=55 time=35.8 ms
64 bytes from 165.22.211.182 (165.22.211.182): icmp_seq=21 ttl=55 time=44.5 ms
```

Use of ipconfig-

It displays all TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, ipconfig displays Internet Protocol Version 4 (IPv4) and (IPv6) addresses.

Use of ifconfig-

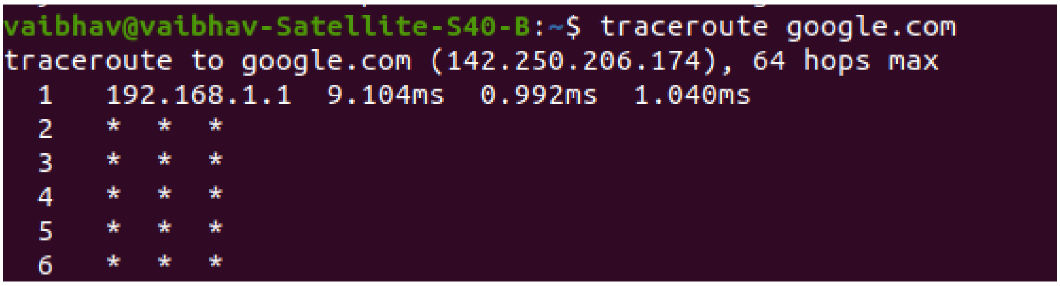
ifconfig stands for Interface Configuration. This command is the same as ipconfig, and is used to view all the current TCP/IP network configurations values of the computer.



```
vaibhav@vaibhav-Satellite-S40-B:~$ ifconfig
enp3s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:97:ba:1a:79:fe  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4042  bytes 419478 (419.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4042  bytes 419478 (419.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.57  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::c348:b75a:b660:6686  prefixlen 64  scopeid 0x20<link>
        ether 30:3a:64:db:79:7d  txqueuelen 1000  (Ethernet)
        RX packets 315585  bytes 363753999 (363.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 105607  bytes 40940084 (40.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vaibhav@vaibhav-Satellite-S40-B:~$
```

Use of tracert/ traceroute-

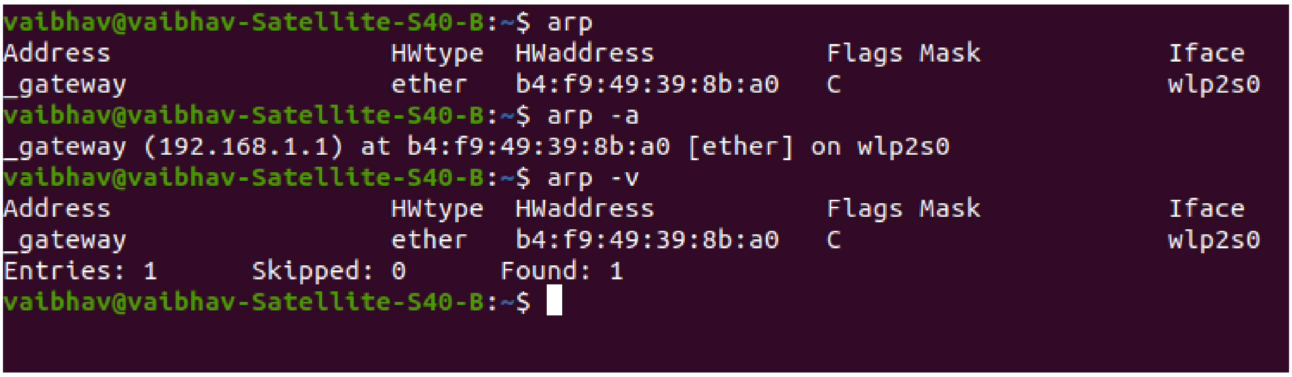
The tracert or traceroute command is a network analysis tool that can be used to know the path a packet goes through or follows from source to destination.



```
vaibhav@vaibhav-Satellite-S40-B:~$ traceroute google.com
traceroute to google.com (142.250.206.174), 64 hops max
  1   192.168.1.1  9.104ms  0.992ms  1.040ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
  5   *  *  *
  6   *  *  *
```

Use of ARP-

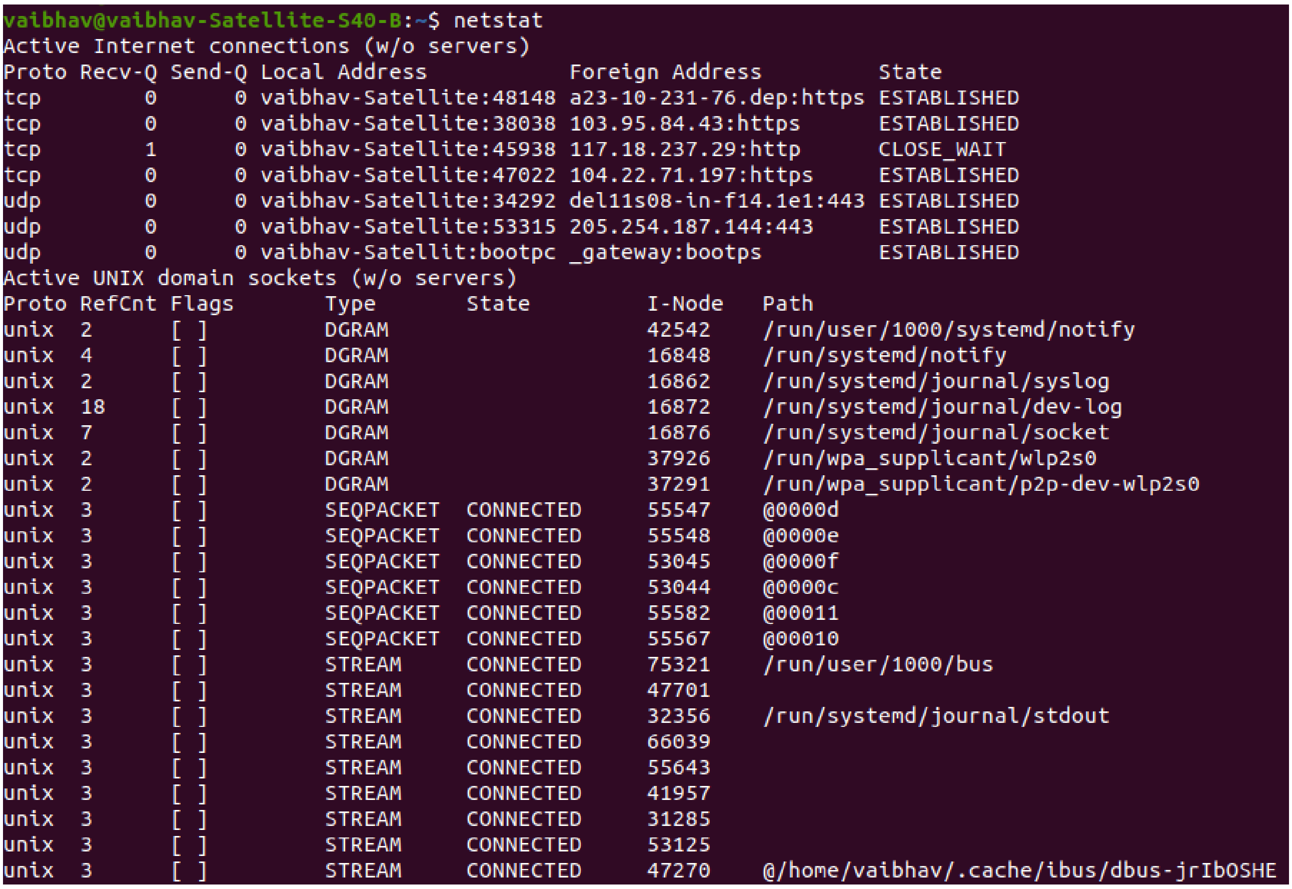
ARP is an Address Resolution Protocol, used to translate between Layer 2 MAC addresses and layer 3 IP addresses. ARP is a program used by a computer system to find another computer’s MAC address based on its IP address.



```
vaibhav@vaibhav-Satellite-S40-B:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   b4:f9:49:39:8b:a0   C                     wlp2s0
vaibhav@vaibhav-Satellite-S40-B:~$ arp -a
_gateway (192.168.1.1) at b4:f9:49:39:8b:a0 [ether] on wlp2s0
vaibhav@vaibhav-Satellite-S40-B:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   b4:f9:49:39:8b:a0   C                     wlp2s0
Entries: 1      Skipped: 0      Found: 1
vaibhav@vaibhav-Satellite-S40-B:~$ 
```

Use of Netstat-

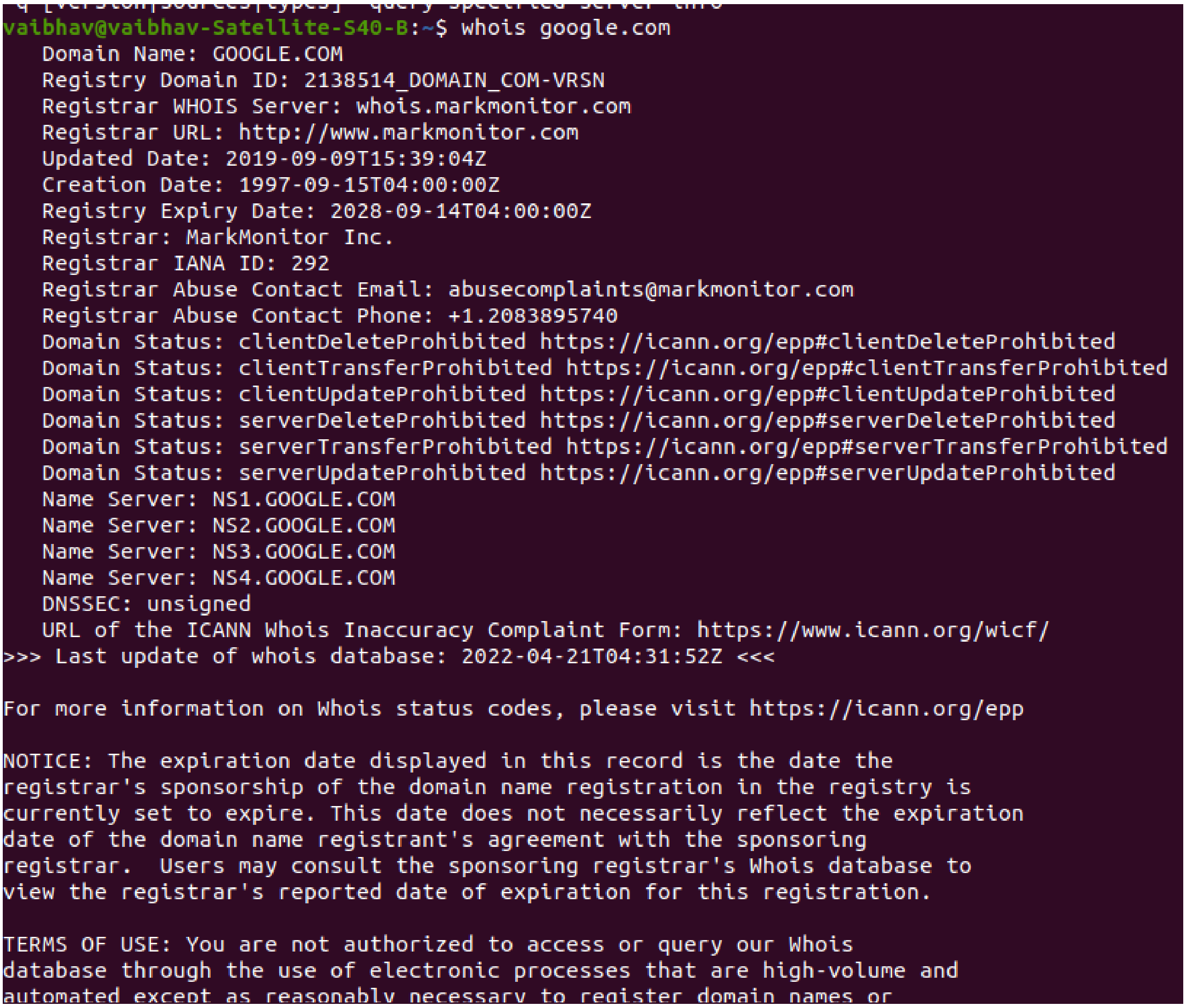
The network statistics (netstat) command is a networking tool used for troubleshooting and configuration, that can also serve as a monitoring tool for connections over the network. Both incoming and outgoing connections, routing tables, port listening and usage statistics are common uses for this command.



```
vaibhav@vaibhav-Satellite-S40-B:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 vaibhav-Satellite:48148 a23-10-231-76.dep:https ESTABLISHED
tcp        0      0 vaibhav-Satellite:38038 103.95.84.43:https      ESTABLISHED
tcp        1      0 vaibhav-Satellite:45938 117.18.237.29:http      CLOSE_WAIT
tcp        0      0 vaibhav-Satellite:47022 104.22.71.197:https     ESTABLISHED
udp        0      0 vaibhav-Satellite:34292 del11s08-in-f14.1e1:443 ESTABLISHED
udp        0      0 vaibhav-Satellite:53315 205.254.187.144:443     ESTABLISHED
udp        0      0 vaibhav-Satellit:bootpc _gateway:bootps         ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    42542    /run/user/1000/systemd/notify
unix  4      [ ]         DGRAM                    16848    /run/systemd/notify
unix  2      [ ]         DGRAM                    16862    /run/systemd/journal/syslog
unix  18     [ ]         DGRAM                    16872    /run/systemd/journal/dev-log
unix  7      [ ]         DGRAM                    16876    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    37926    /run/wpa_supplicant/wlp2s0
unix  2      [ ]         DGRAM                    37291    /run/wpa_supplicant/p2p-dev-wlp2s0
unix  3      [ ]         SEQPACKET  CONNECTED     55547    @0000d
unix  3      [ ]         SEQPACKET  CONNECTED     55548    @0000e
unix  3      [ ]         SEQPACKET  CONNECTED     53045    @0000f
unix  3      [ ]         SEQPACKET  CONNECTED     53044    @0000c
unix  3      [ ]         SEQPACKET  CONNECTED     55582    @00011
unix  3      [ ]         SEQPACKET  CONNECTED     55567    @00010
unix  3      [ ]         STREAM     CONNECTED     75321    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     47701
unix  3      [ ]         STREAM     CONNECTED     32356    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     66039
unix  3      [ ]         STREAM     CONNECTED     55643
unix  3      [ ]         STREAM     CONNECTED     41957
unix  3      [ ]         STREAM     CONNECTED     31285
unix  3      [ ]         STREAM     CONNECTED     53125
unix  3      [ ]         STREAM     CONNECTED     47270    @/home/vaibhav/.cache/ibus/dbus-jrIbOSHE
```

Use of WHOIS command-

WHOIS is a query and response protocol that is widely used for querying databases that store the registered users of an Internet resource, such as a domain name or an IP address block, but is also used for a wider range of other information.



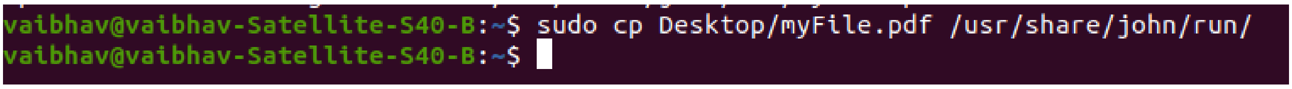
```
-q [version|sources|types]  query specified server info
vaibhav@vaibhav-Satellite-S40-B:~$ whois google.com
   Domain Name: GOOGLE.COM
   Registry Domain ID: 2138514_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-04-21T04:31:52Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar.  Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
```

2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.

John the Ripper is a password cracking software tool. Originally developed for the Unix operating system, it can run on fifteen different (eleven of which are architecture-specific versions of Unix, DOS, Win32, BeOS and OpenVMS). It is among the most frequently used password testing and breaking programs as it combines a number of password crackers into one package, autodetects password hash types and includes a customizable cracker. It can be run against various encrypted password formats including several crypt password hash types most commonly found on various Unix version (based on DES, MD5 or Blowfish), Kerberos AFS and Windows version. Additional modules have extended its ability to include MD4-based password hashes and passwords stores in LDAP, MySQL and others.

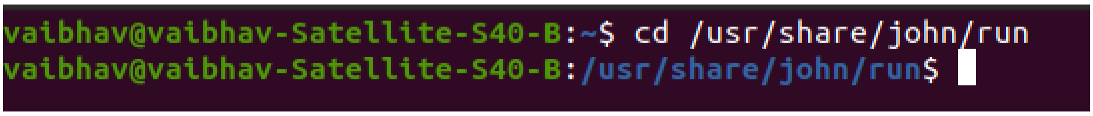
There are few steps to decrypt the passsword protected file......

1. First we will create a password protected file(PDF).
2. Then we will send that file to the john-the-ripper/run folder.



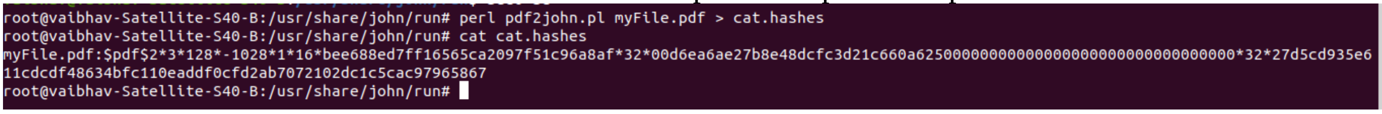
```
vaibhav@vaibhav-Satellite-S40-B:~$ sudo cp Desktop/myFile.pdf /usr/share/john/run/
vaibhav@vaibhav-Satellite-S40-B:~$ 
```

3. Now we’ll jump to the folder where we copied our password file.



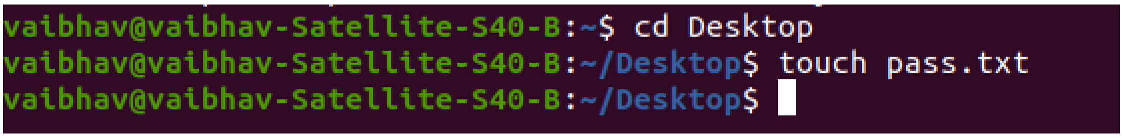
```
vaibhav@vaibhav-Satellite-S40-B:~$ cd /usr/share/john/run
vaibhav@vaibhav-Satellite-S40-B:/usr/share/john/run$ 
```

4. In this folder we will create the hash file of our password protected pdf file.



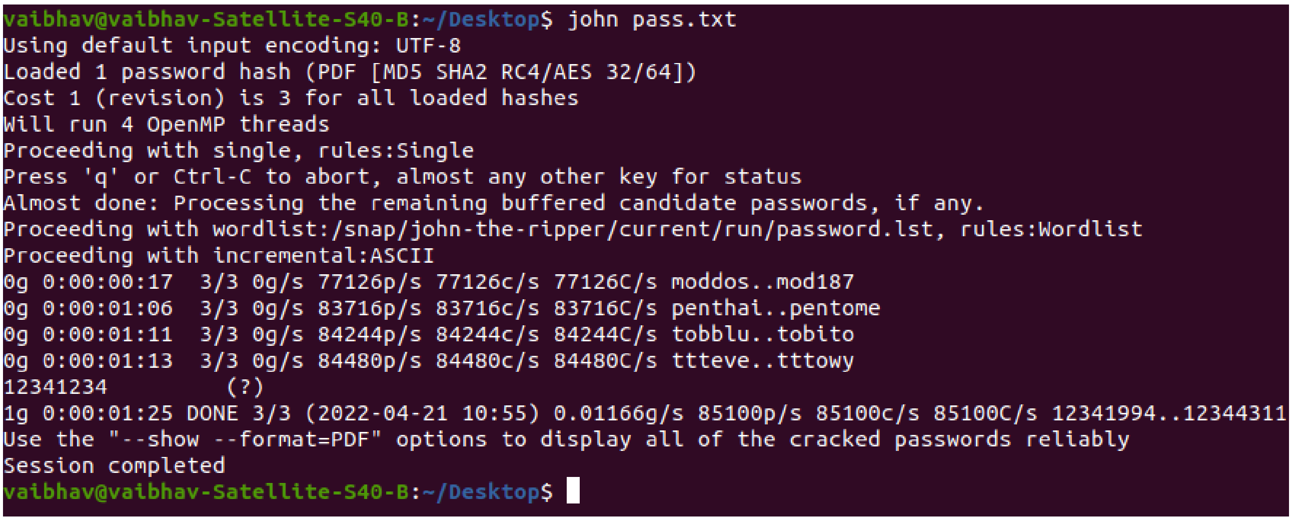
```
root@vaibhav-Satellite-S40-B:/usr/share/john/run# perl pdf2john.pl myFile.pdf > cat.hashes
root@vaibhav-Satellite-S40-B:/usr/share/john/run# cat cat.hashes
myFile.pdf:$pdf$2*3*128*-1028*1*16*bee688ed7ff16565ca2097f51c96a8af*32*00d6ea6ae27b8e48dcfc3d21c660a62500000000000000000000000000000000*32*27d5cd935e6
11cdcdf48634bfc110eaddf0cfd2ab7072102dc1c5cac97965867
root@vaibhav-Satellite-S40-B:/usr/share/john/run# 
```

5. Now our pdf file password has been converted to has hash file, we have to save these hase codes into a txt file.



```
vaibhav@vaibhav-Satellite-S40-B:~$ cd Desktop
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ touch pass.txt
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ 
```

6. We have to use john to crack the hash code of our file.

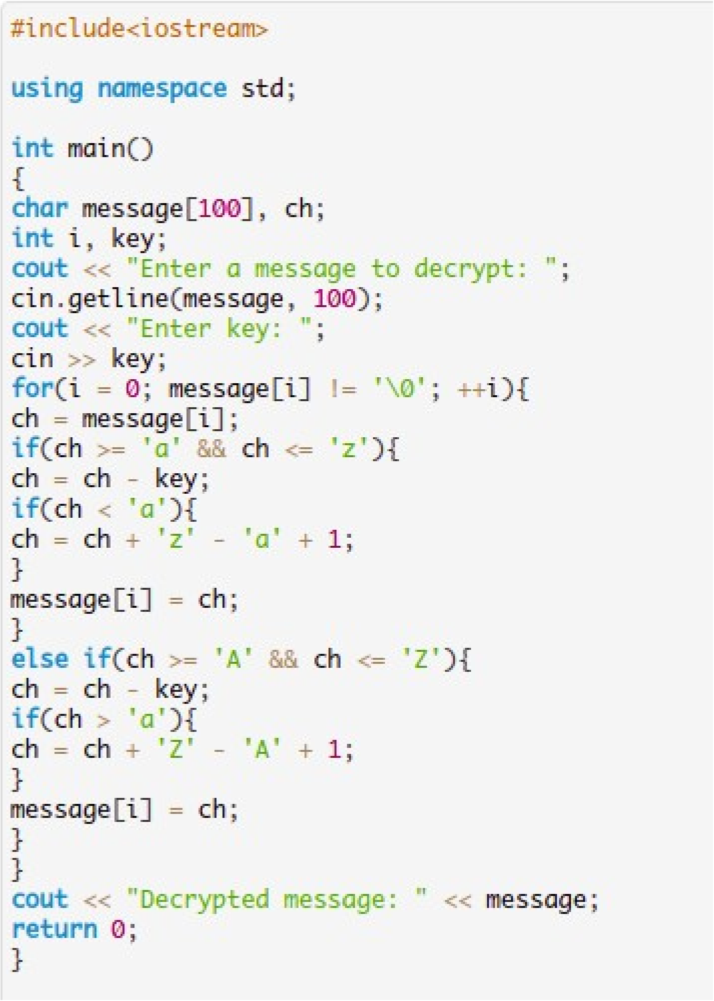


```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ john pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Cost 1 (revision) is 3 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/snap/john-the-ripper/current/run/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:17  3/3 0g/s 77126p/s 77126c/s 77126C/s moddos..mod187
0g 0:00:01:06  3/3 0g/s 83716p/s 83716c/s 83716C/s penthai..pentome
0g 0:00:01:11  3/3 0g/s 84244p/s 84244c/s 84244C/s tobblu..tobito
0g 0:00:01:13  3/3 0g/s 84480p/s 84480c/s 84480C/s ttteve..tttowy
12341234         (?)
1g 0:00:01:25 DONE 3/3 (2022-04-21 10:55) 0.01166g/s 85100p/s 85100c/s 85100C/s 12341994..12344311
Use the "--show --format=PDF" options to display all of the cracked passwords reliably
Session completed
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ 
```

3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.

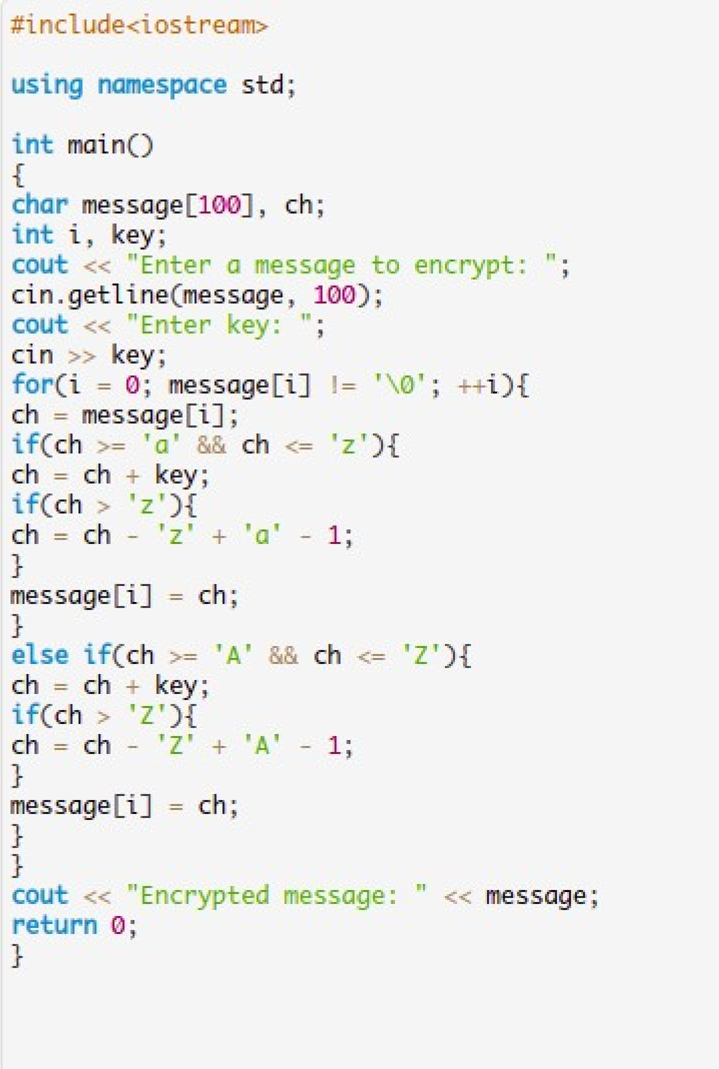
A Caesar cipher is a simple method of encoding messages. Caesar ciphers use a substitution method where letters in the alphabet are shifted by some fixed number of spaces to yield an encoding alphabet. A Caesar cipher with a shift of 1 would encode an A as a B, an M as an N, and a Z as an A, and so on.

This c++ code is for encryption....



```
#include<iostream>

using namespace std;

int main()
{
char message[100], ch;
int i, key;
cout << "Enter a message to decrypt: ";
cin.getline(message, 100);
cout << "Enter key: ";
cin >> key;
for(i = 0; message[i] != '\0'; ++i){
ch = message[i];
if(ch >= 'a' && ch <= 'z'){
ch = ch - key;
if(ch < 'a'){
ch = ch + 'z' - 'a' + 1;
}
message[i] = ch;
}
else if(ch >= 'A' && ch <= 'Z'){
ch = ch - key;
if(ch > 'a'){
ch = ch + 'Z' - 'A' + 1;
}
message[i] = ch;
}
}
cout << "Decrypted message: " << message;
return 0;
}
```

And this c++ code is for decryption...



```
#include<iostream>

using namespace std;

int main()
{
char message[100], ch;
int i, key;
cout << "Enter a message to encrypt: ";
cin.getline(message, 100);
cout << "Enter key: ";
cin >> key;
for(i = 0; message[i] != '\0'; ++i){
ch = message[i];
if(ch >= 'a' && ch <= 'z'){
ch = ch + key;
if(ch > 'z'){
ch = ch - 'z' + 'a' - 1;
}
message[i] = ch;
}
else if(ch >= 'A' && ch <= 'Z'){
ch = ch + key;
if(ch > 'Z'){
ch = ch - 'Z' + 'A' - 1;
}
message[i] = ch;
}
}
cout << "Encrypted message: " << message;
return 0;
}
```

4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.

The rail fence cipher is a form of transposition cipher. It derives its name from the way in which it is encoded.

```
// C++ program to illustrate Rail Fence Cipher
// Encryption and Decryption
#include <bits/stdc++.h>
using namespace std;

// function to encrypt a message
string encryptRailFence(string text, int key)
{
        // create the matrix to cipher plain text
        // key = rows , length(text) = columns
        char rail[key][(text.length())];
```

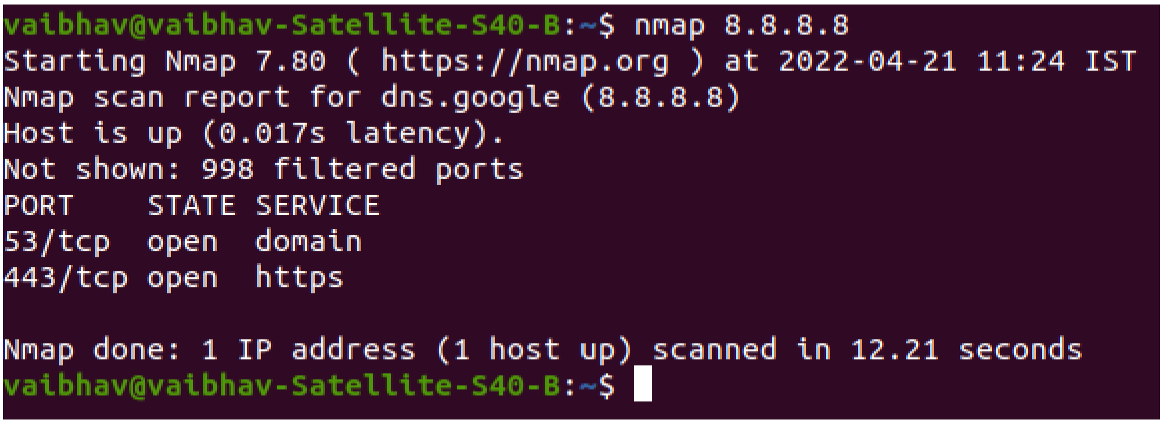
```
	// filling the rail matrix to distinguish filled
	// spaces from blank ones
	for (int i=0; i < key; i++)
		for (int j = 0; j < text.length(); j++)
			rail[i][j] = '\n';

	// to find the direction
	bool dir_down = false;
	int row = 0, col = 0;

	for (int i=0; i < text.length(); i++)
	{
		// check the direction of flow
		// reverse the direction if we've just
		// filled the top or bottom rail
		if (row == 0 || row == key-1)
			dir_down = !dir_down;

		// fill the corresponding alphabet
		rail[row][col++] = text[i];

		// find the next row using direction flag
		dir_down?row++ : row--;
	}

	//now we can construct the cipher using the rail matrix
	string result;
	for (int i=0; i < key; i++)
		for (int j=0; j < text.length(); j++)
			if (rail[i][j]!='\n')
				result.push_back(rail[i][j]);

	return result;
}

// This function receives cipher-text and key
// and returns the original text after decryption
string decryptRailFence(string cipher, int key)
{
	// create the matrix to cipher plain text
	// key = rows , length(text) = columns
	char rail[key][cipher.length()];

	// filling the rail matrix to distinguish filled
	// spaces from blank ones
	for (int i=0; i < key; i++)
		for (int j=0; j < cipher.length(); j++)
			rail[i][j] = '\n';

	// to find the direction
```

```
        bool dir_down;

        int row = 0, col = 0;

        // mark the places with '*'
        for (int i=0; i < cipher.length(); i++)
        {
                // check the direction of flow
                if (row == 0)
                        dir_down = true;
                if (row == key-1)
                        dir_down = false;

                // place the marker
                rail[row][col++] = '*';

                // find the next row using direction flag
                dir_down?row++ : row--;
        }

        // now we can construct the fill the rail matrix
        int index = 0;
        for (int i=0; i<key; i++)
                for (int j=0; j<cipher.length(); j++)
                        if (rail[i][j] == '*' && index<cipher.length())
                                rail[i][j] = cipher[index++];


        // now read the matrix in zig-zag manner to construct
        // the resultant text
        string result;

        row = 0, col = 0;
        for (int i=0; i< cipher.length(); i++)
        {
                // check the direction of flow
                if (row == 0)
                        dir_down = true;
                if (row == key-1)
                        dir_down = false;

                // place the marker
                if (rail[row][col] != '*')
                        result.push_back(rail[row][col++]);

                // find the next row using direction flag
                dir_down?row++: row--;
        }
        return result;
}
```

```
//driver program to check the above functions
int main()
{
	cout << encryptRailFence("Hello there", 2) << endl;

	//Now decryption of the same cipher-text
	cout << decryptRailFence("",2) << endl;
	return 0;
}
```

5. Use nmap/zenmap to analyse a remote machine.

Nmap allows us to scan our network and discover not only everything connected to it, but also a wide variety of information about what's connected, what services each host is operating, and so on. It allows a large number of scanning techniques, such as UDP, TCP connect (), TCP SYN and FTP.



```
vaibhav@vaibhav-Satellite-S40-B:~$ nmap 8.8.8.8
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-21 11:24 IST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.017s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
53/tcp  open  domain
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 12.21 seconds
vaibhav@vaibhav-Satellite-S40-B:~$ 
```

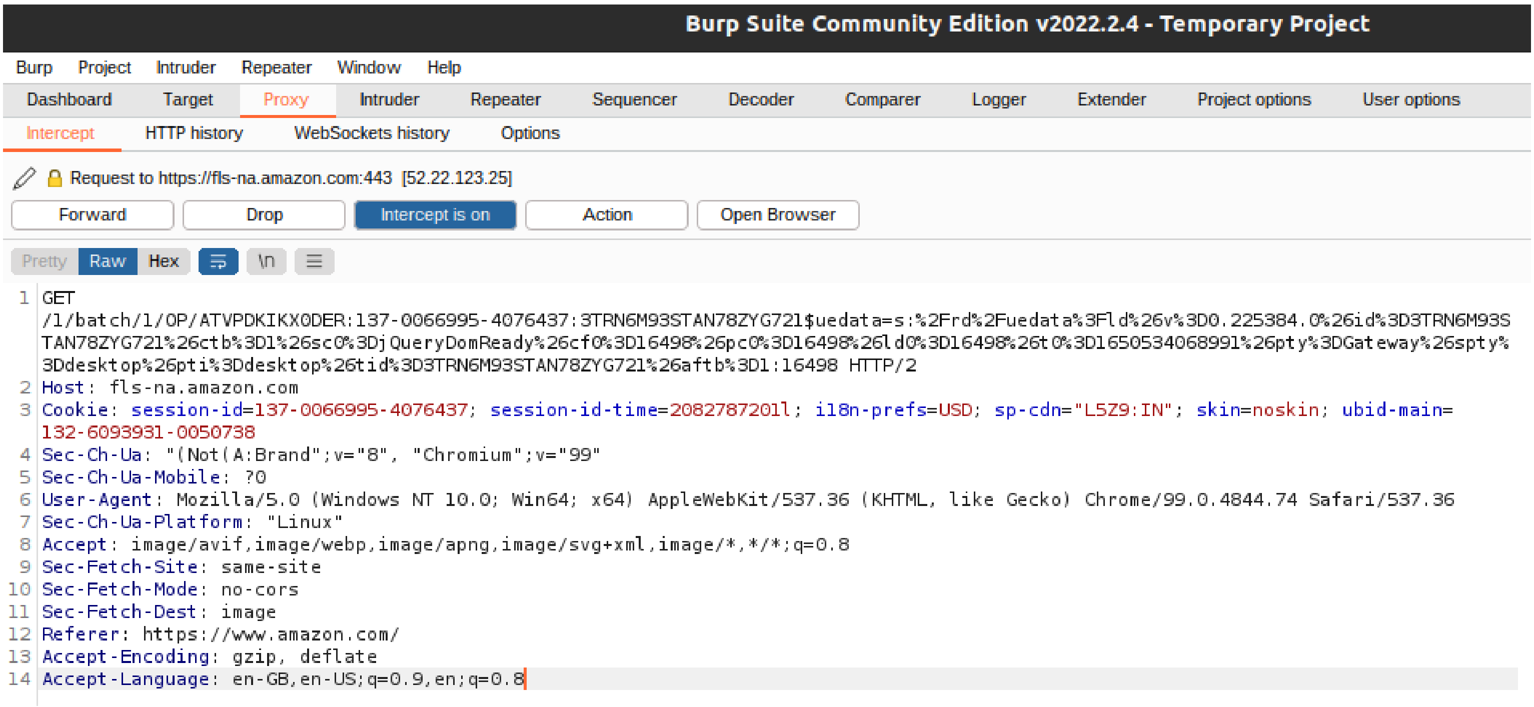
6. Use Burp proxy to capture and modify the message.

Burp suite is an integrated platform/graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

i) First we will install the updated version of java using the command-
sudo apt-get install openjdk-8-jre

ii) Install burp_suite community edition from their website.
iii) Change the permission by
chmod u+x (burp_suite_file.sh)

iv) Run the file.


Burp Suite Community Edition v2022.2.4 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Open Browser
Pretty Raw Hex \n

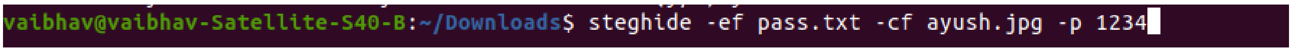
Request to https://fls-na.amazon.com:443 [52.22.123.25]

```
GET
/1/batch/1/OP/ATVPDKIKX0DER:137-0066995-4076437:3TRN6M93STAN78ZYG721$uedata=s:%2Frd%2Fuedata%3Fld%26v%3D0.225384.0%26id%3D3TRN6M93S
TAN78ZYG721%26ctb%3D1%26sc0%3DjQueryDomReady%26cf0%3D16498%26pc0%3D16498%26ld0%3D16498%26t0%3D1650534068991%26pty%3DGateway%26spty%
3Ddesktop%26pti%3Ddesktop%26tid%3D3TRN6M93STAN78ZYG721%26aftb%3D1:16498 HTTP/2
Host: fls-na.amazon.com
Cookie: session-id=137-0066995-4076437; session-id-time=2082787201l; i18n-prefs=USD; sp-cdn="L5Z9:IN"; skin=noskin; ubid-main=
132-6093931-0050738
Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="99"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: https://www.amazon.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
```

Intercept in proxy can pause the loading of the page which is opened and the data can easily be changed by burp suite and Forward button processes the loading the page.

7. Demonstrate the use of steganography tools.

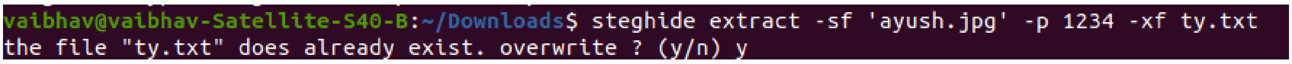
i) First install the steghide using linux terminal.
ii) Write the following command.



```
vaibhav@vaibhav-Satellite-S40-B:~/Downloads$ steghide -ef pass.txt -cf ayush.jpg -p 1234
```

Here we are calling steghide to embedd file (-ef) to a cover file (-cf) with password as 1234 (-p)

iii) As the file has been embedded to the image now we have to transfer the data of the hidden file to another file. Using the following command.

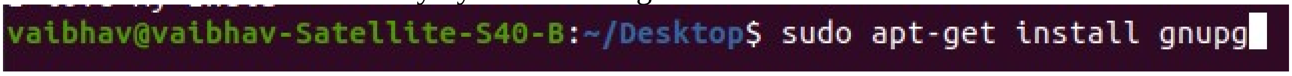


```
vaibhav@vaibhav-Satellite-S40-B:~/Downloads$ steghide extract -sf 'ayush.jpg' -p 1234 -xf ty.txt
the file "ty.txt" does already exist. overwrite ? (y/n) y
```

8. Demonstrate use of gpg utility for signing and encrypting purposes.

GPG- GNU Private Guard, is a public key cryptography implementation. This allows the secure transmission of information between parties and can be used to verify that the origin of a message is genuine.
1. GPG for encrypting purpose..

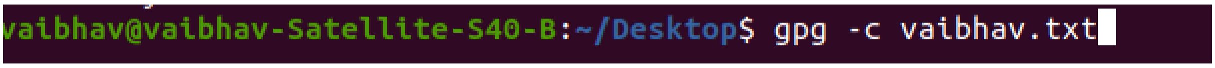
i. First we will install GPG utility by the following command..



```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ sudo apt-get install gnupg
```

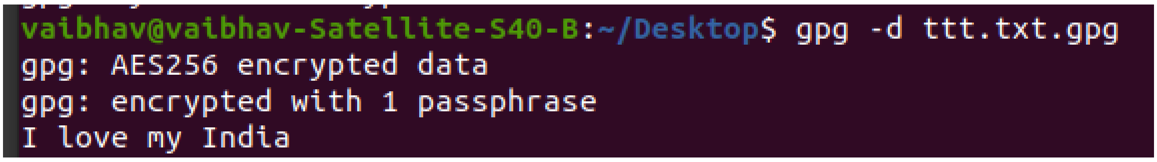
ii. After installing we have to create a text file can be named as vaibhav.txt by the command..
touch vaibhav.txt

iii. If we wanna encrypt this file by GPG following command will be used.



```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ gpg -c vaibhav.txt
```

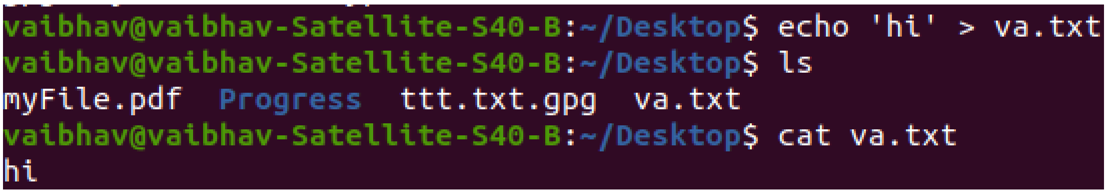
iv. If we wanna decrypt the data we will follow the command..



```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ gpg -d ttt.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
I love my India
```

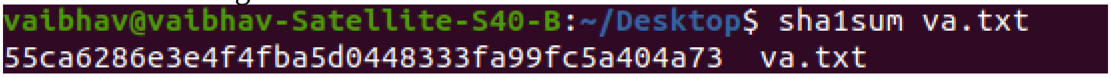
2. GPG utility for signing purposes...

i. For this again we will create a txt file, but now using a different command



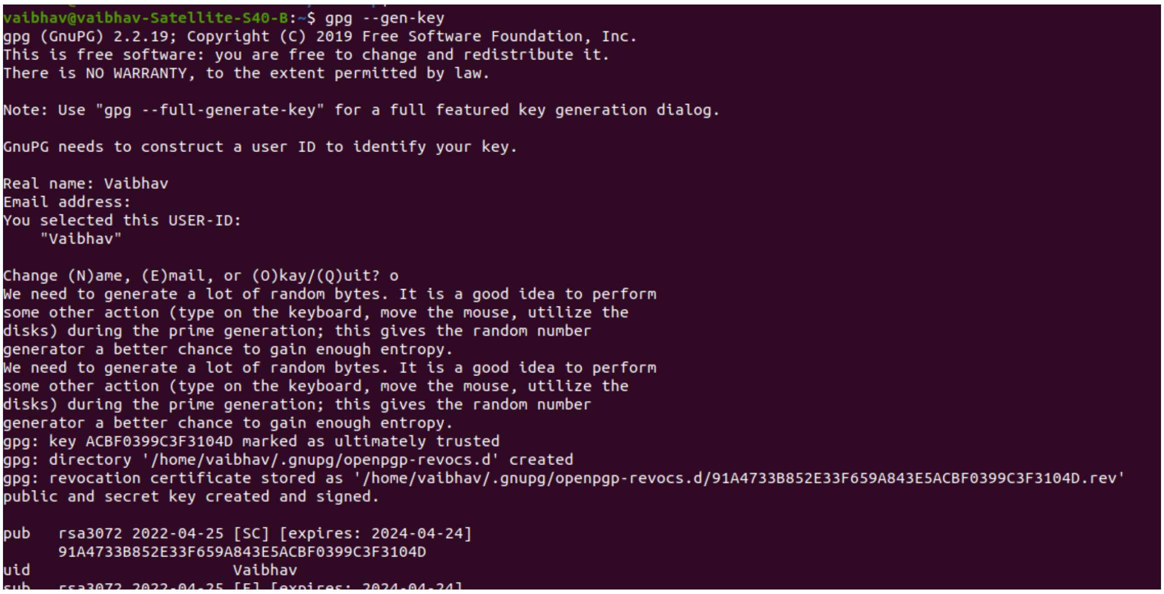
```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ echo 'hi' > va.txt
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ ls
myFile.pdf  Progress  ttt.txt.gpg  va.txt
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ cat va.txt
hi
```

ii. Now will will generate the hash of this file.



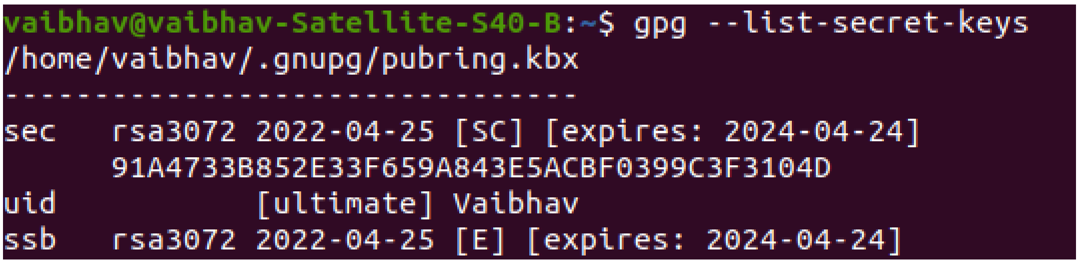
```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ sha1sum va.txt
55ca6286e3e4f4fba5d0448333fa99fc5a404a73  va.txt
```

iii. After having the hash code will have to generate some GPG keys.



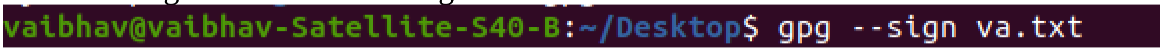
```
vaibhav@vaibhav-Satellite-S40-B:~$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Vaibhav
Email address:
You selected this USER-ID:
    "Vaibhav"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key ACBF0399C3F3104D marked as ultimately trusted
gpg: directory '/home/vaibhav/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/vaibhav/.gnupg/openpgp-revocs.d/91A4733B852E33F659A843E5ACBF0399C3F3104D.rev'
public and secret key created and signed.

pub   rsa3072 2022-04-25 [SC] [expires: 2024-04-24]
      91A4733B852E33F659A843E5ACBF0399C3F3104D
uid                      Vaibhav
sub   rsa3072 2022-04-25 [E] [expires: 2024-04-24]
```

iv. Now we have our keys. We can take a look at our secret keys.. by using the following command.



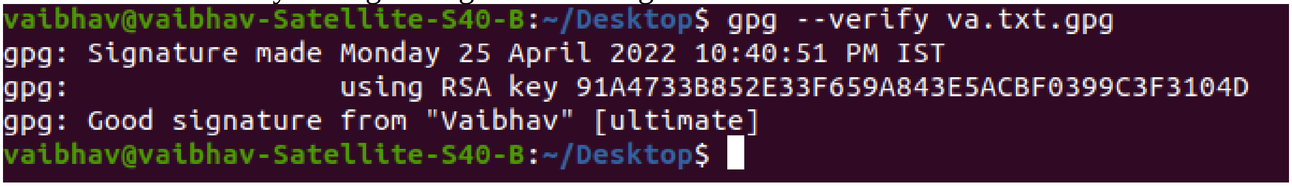
```
vaibhav@vaibhav-Satellite-S40-B:~$ gpg --list-secret-keys
/home/vaibhav/.gnupg/pubring.kbx
--------------------------------
sec   rsa3072 2022-04-25 [SC] [expires: 2024-04-24]
      91A4733B852E33F659A843E5ACBF0399C3F3104D
uid           [ultimate] Vaibhav
ssb   rsa3072 2022-04-25 [E] [expires: 2024-04-24]
```

v. Now we will sign the document using GPG



```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ gpg --sign va.txt
```

vi. Now we can verify the sign using the following command...



```
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ gpg --verify va.txt.gpg
gpg: Signature made Monday 25 April 2022 10:40:51 PM IST
gpg:                using RSA key 91A4733B852E33F659A843E5ACBF0399C3F3104D
gpg: Good signature from "Vaibhav" [ultimate]
vaibhav@vaibhav-Satellite-S40-B:~/Desktop$ 
```