Assignment-1

Aim: Shift cipher and mono alphabet substitution cipher

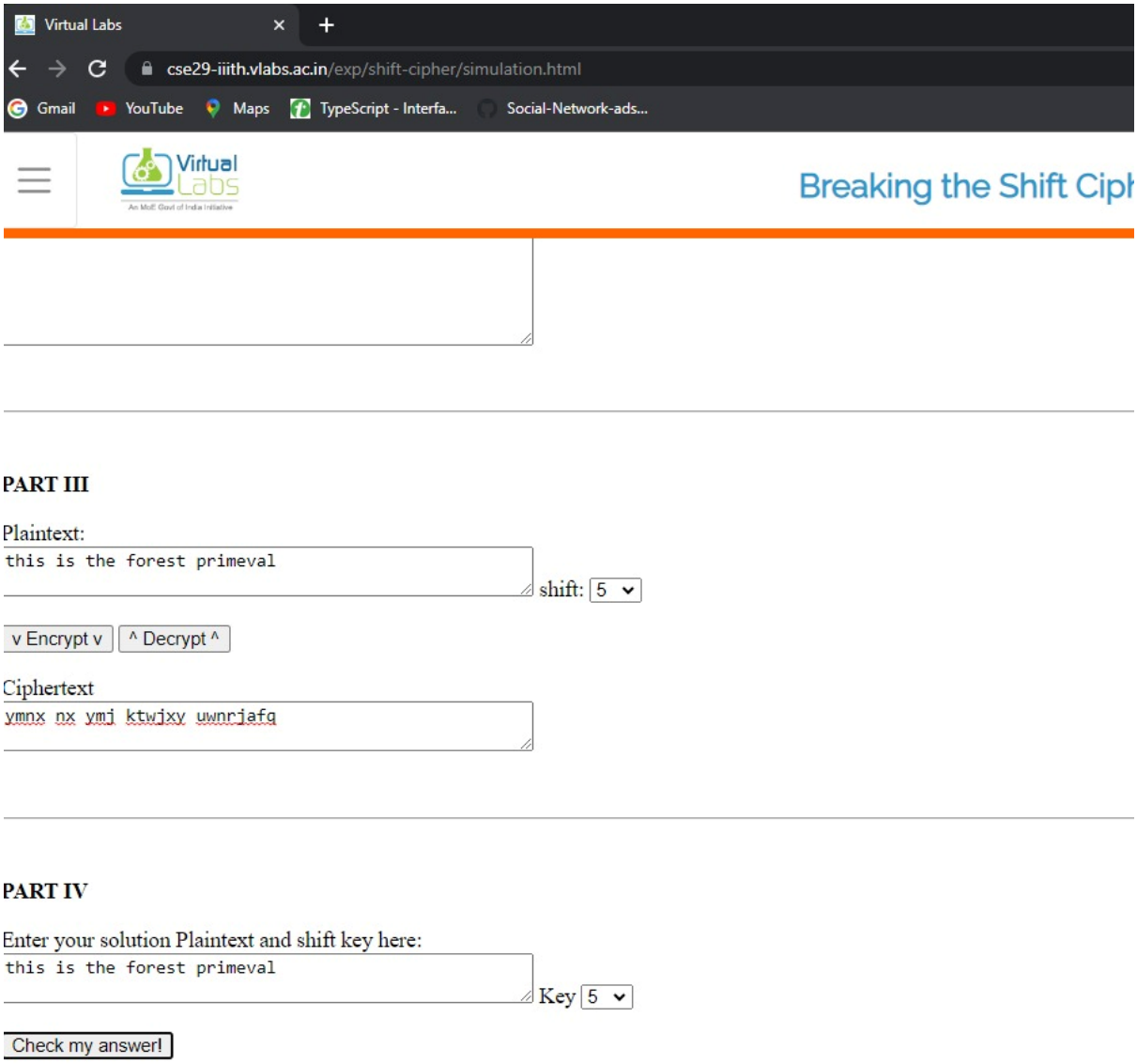
Theory:

Shift cipher:- The Caesar cipher is a simple encryption technique that was used by Julius Caesar to send secret messages to his allies. It works by shifting the letters in the plaintext message by a certain number of positions, known as the “shift” or “key” cipher.

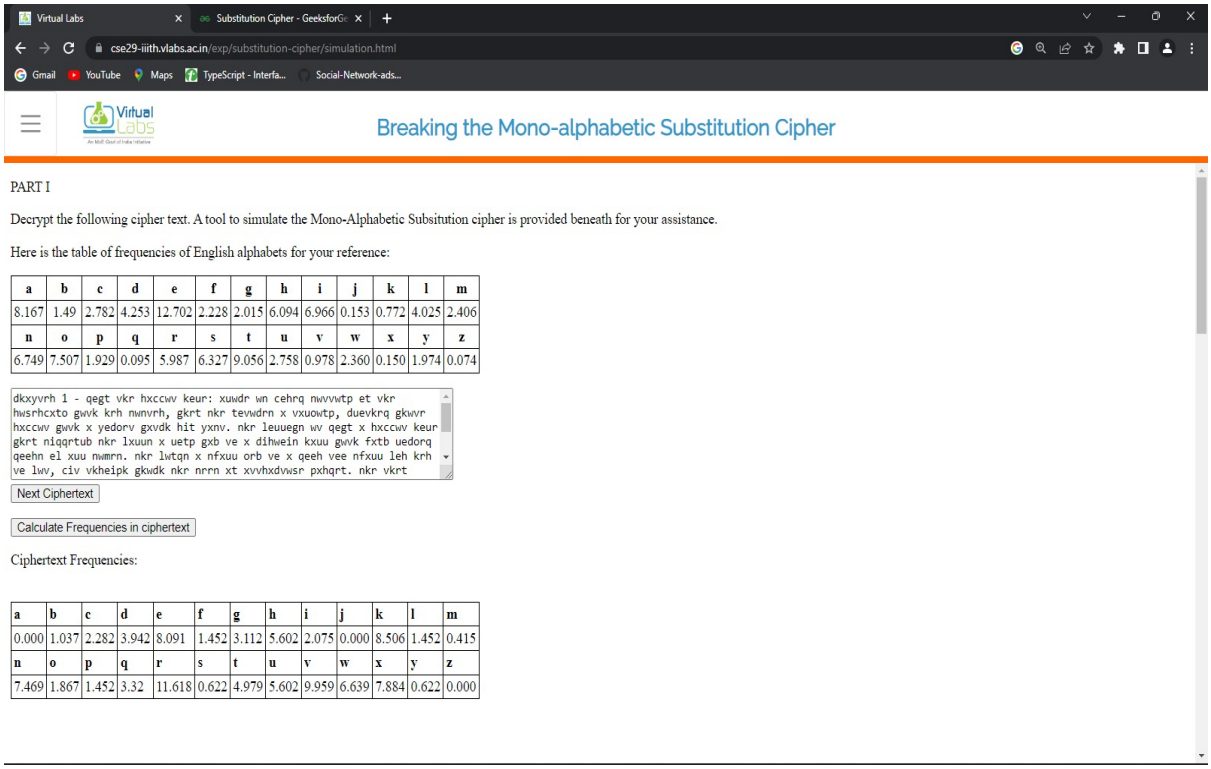
The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.
Thus to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down.

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,..., Z = 25. Encryption of a letter by a shift n can be described mathematically as.

For example, if the shift is 3, then the letter A would be replaced by the letter D, B would become E, C would become F, and so on. The alphabet is wrapped around so that after Z, it starts back at A.



Mono alphabet substitution cipher:- It refers to that cipher in which all the letters of the plain text get mapped into the cipher text letters, on the basis of one alphabetic key. The Caesar-shift cipher is one of the major examples of a monoalphabetic cipher. Here, every letter shifts on the basis of a numeric key. The atbash cipher is another example. Here, every letter gets mapped to its symmetric letter (symmetric to the alphabet's centre).



PART I

Decrypt the following cipher text. A tool to simulate the Mono-Alphabetic Subsitution cipher is provided beneath for your assistance.

Here is the table of frequencies of English alphabets for your reference:

a	b	c	d	e	f	g	h	i	j	k	l	m
8.167	1.49	2.782	4.253	12.702	2.228	2.015	6.094	6.966	0.153	0.772	4.025	2.406
n	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
6.749	7.507	1.929	0.095	5.987	6.327	9.056	2.758	0.978	2.360	0.150	1.974	0.074

```
dkxyvrh 1 - qegt vkr hxccwv keur: xuwdr wn cehrq nwvvwtp et vkr
hwsrhcxto gwvk krh nwnvrh, gkrt nkr tevwdrn x vxuowtp, duevkrq gkwvr
hxccwv gwvk x yedorv gxvdk hit yxnv. nkr leuuegn wv qegt x hxccwv keur
gkrt niqqrtub nkr lxuun x uetp gxb ve x dihwein kxuu gwvk fxtb uedorq
qeehn el xuu nwmrn. nkr lwtqn x nfxuu orb ve x qeeh vee nfxuu leh krh
ve lwv, civ vkheipk gkwdk nkr nrrn xt xvvhxdvwsr pxhqrt. nkr vkrt
```

Next Ciphertext

Calculate Frequencies in ciphertext

Ciphertext Frequencies:

a	b	c	d	e	f	g	h	i	j	k	l	m
0.000	1.037	2.282	3.942	8.091	1.452	3.112	5.602	2.075	0.000	8.506	1.452	0.415
n	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
7.469	1.867	1.452	3.32	11.618	0.622	4.979	5.602	9.959	6.639	7.884	0.622	0.000

replaced y by P You replaced h by R You replaced w by I You replaced g by W You replaced o by K You replaced q by D You replaced n by S You replaced p by G You replaced s by V You replaced c by B You replaced i by U You replaced u by L You replaced f by M You replaced m by Z You replaced l by F You replaced b by Y

PART III

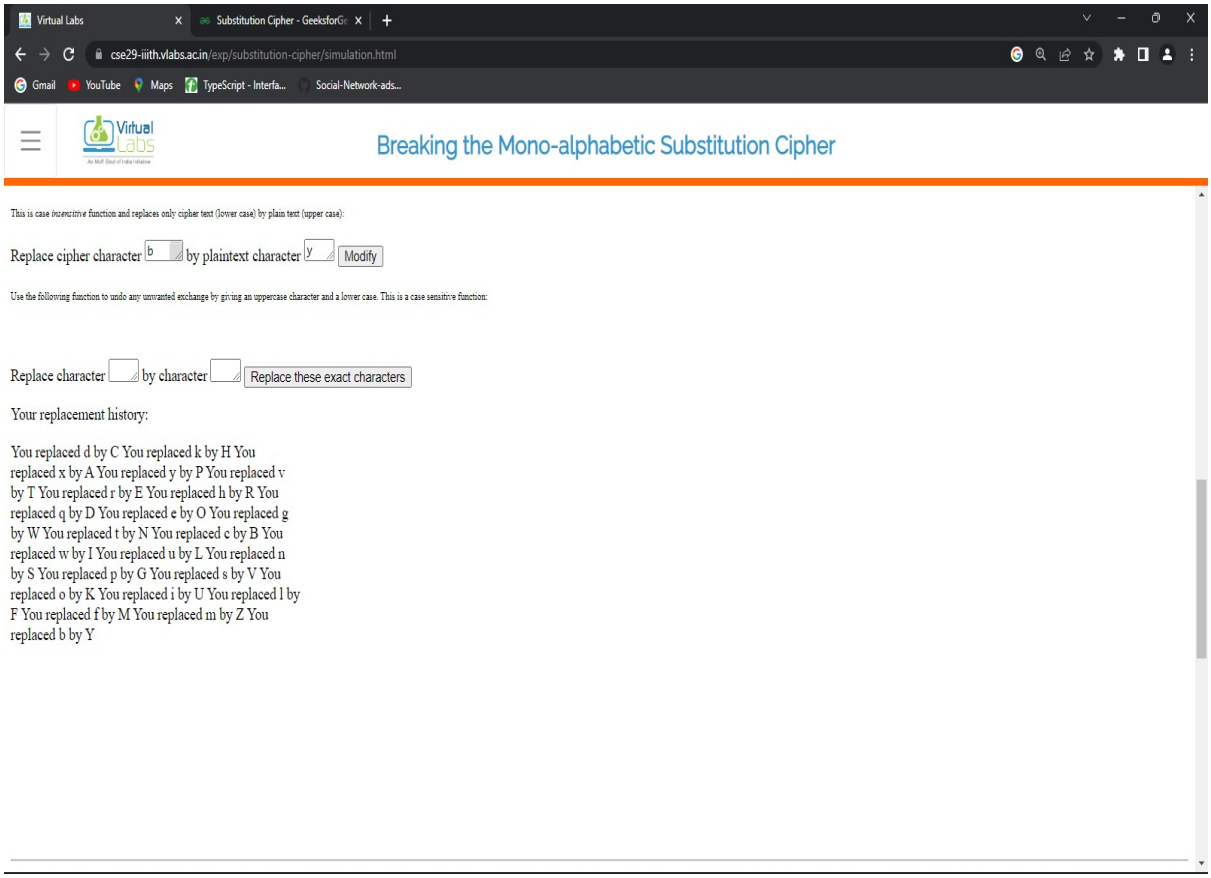
Enter your solution plaintext here:

```
CHAPTER 1 - DOWN THE RABBIT HOLE: ALICE IS BORED SITTING ON THE
RIVERBANK WITH HER SISTER, WHEN SHE NOTICES A TALKING, CLOTHED WHITE
RABBIT WITH A POCKET WATCH RUN PAST. SHE FOLLOWS IT DOWN A RABBIT HOLE
WHEN SUDDENLY SHE FALLS A LONG WAY TO A CURIOUS HALL WITH MANY LOCKED
DOORS OF ALL SIZES. SHE FINDS A SMALL KEY TO A DOOR TOO SMALL FOR HER
```

Solution Key = xcdqrlpkwzoufteyahnvisgjbm

Check Answer!

```
CORRECT!!
```


Virtual Labs
Substitution Cipher - GeeksforGe
cse29-iiith.vlabs.ac.in/exp/substitution-cipher/simulation.html
Gmail
YouTube
Maps
TypeScript - Interfa...
Social-Network-ads...
Virtual Labs
Breaking the Mono-alphabetic Substitution Cipher
This is case insensitive function and replaces only cipher text (lower case) by plain text (upper case):
Replace cipher character b by plaintext character y Modify
Use the following function to undo any unwanted exchange by giving an uppercase character and a lower case. This is a case sensitive function:
Replace character by character Replace these exact characters
Your replacement history:
You replaced d by C You replaced k by H You replaced x by A You replaced y by P You replaced v by T You replaced r by E You replaced h by R You replaced q by D You replaced e by O You replaced g by W You replaced t by N You replaced c by B You replaced w by I You replaced u by L You replaced n by S You replaced p by G You replaced s by V You replaced o by K You replaced i by U You replaced l by F You replaced f by M You replaced m by Z You replaced b by Y

Experiment 02 Moheet Shendarkar

Date: Roll No: 122

Aim: To study and perform implementation of Playfair and Vigenere cipher.

Theory:

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike **traditional cipher** we encrypt a pair of alphabets(digraphs) instead of a single alphabet.

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

1. **Generate the key Square(5×5):**
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.

   - The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

2. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.
   For example:

```
PlainText: "instruments"
After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'
```

1. Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.
Plain Text: “hello”
After Split: ‘he’ ‘lx’ ‘lo’
Here **‘x’** is the bogus letter.
2. If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter
Plain Text: “helloe”
AfterSplit: ‘he’ ‘lx’ ‘lo’ ‘ez’

Here **‘z’** is the bogus letter.
Rules for Encryption:

- **If both the letters are in the same column**: Take the letter below each one (going back to the top if at the bottom).
  For example:

```
Diagraph: "me"
Encrypted Text: cl
Encryption:
  m -> c
  e -> l
```

- **If both the letters are in the same row**: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
  For example:

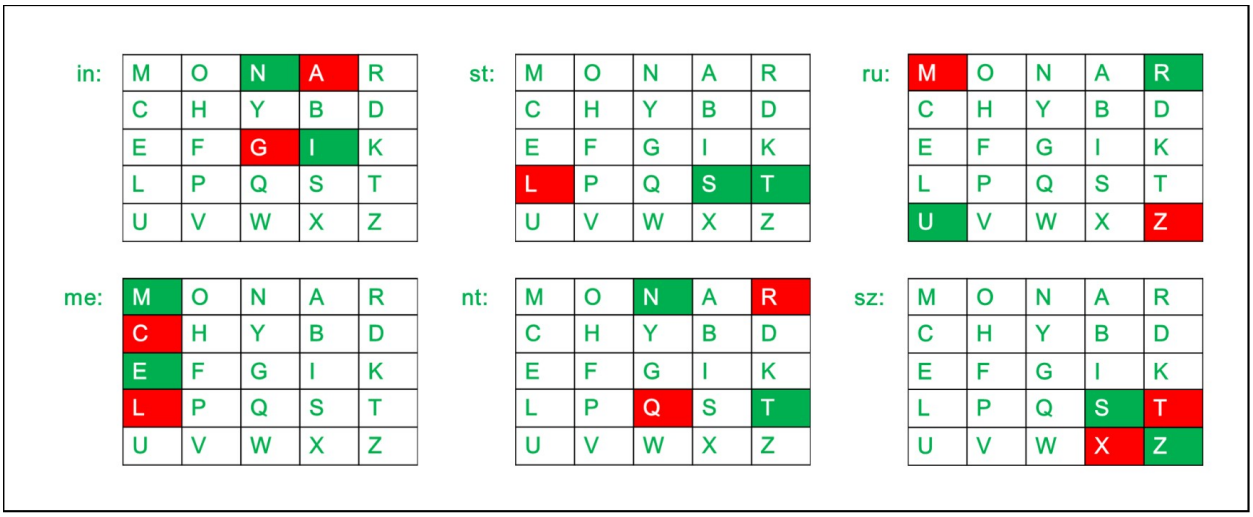
```
Diagraph: "st"
Encrypted Text: tl
Encryption:
  s -> t
  t -> l
```

- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.
  For example:

```
Diagraph: "nt"
Encrypted Text: rq
Encryption:
  n -> r
  t -> q
```

For example:

```
Plain Text: "instrumentsz"
Encrypted Text: gatlmzclrqtx
```



Vigenere Cipher:

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the *Vigenère square or Vigenère table*.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

Encryption:

The first letter of the plaintext, G is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E, and column Y is C. The rest of the plaintext is enciphered in a similar fashion.

Decryption:

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter. Next, we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.

A more **easy implementation** could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

`Encryption`

`The plaintext(P) and key(K) are added modulo 26.`

$E_i = (P_i + K_i) \bmod 26$

Decryption
$D_i = (E_i - K_i) \bmod 26$

Example:
Input : Plaintext : GEEKSFORGEEKS

Keyword : AYUSH

Output : Ciphertext : GCYCZFMLYLEIM

For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.

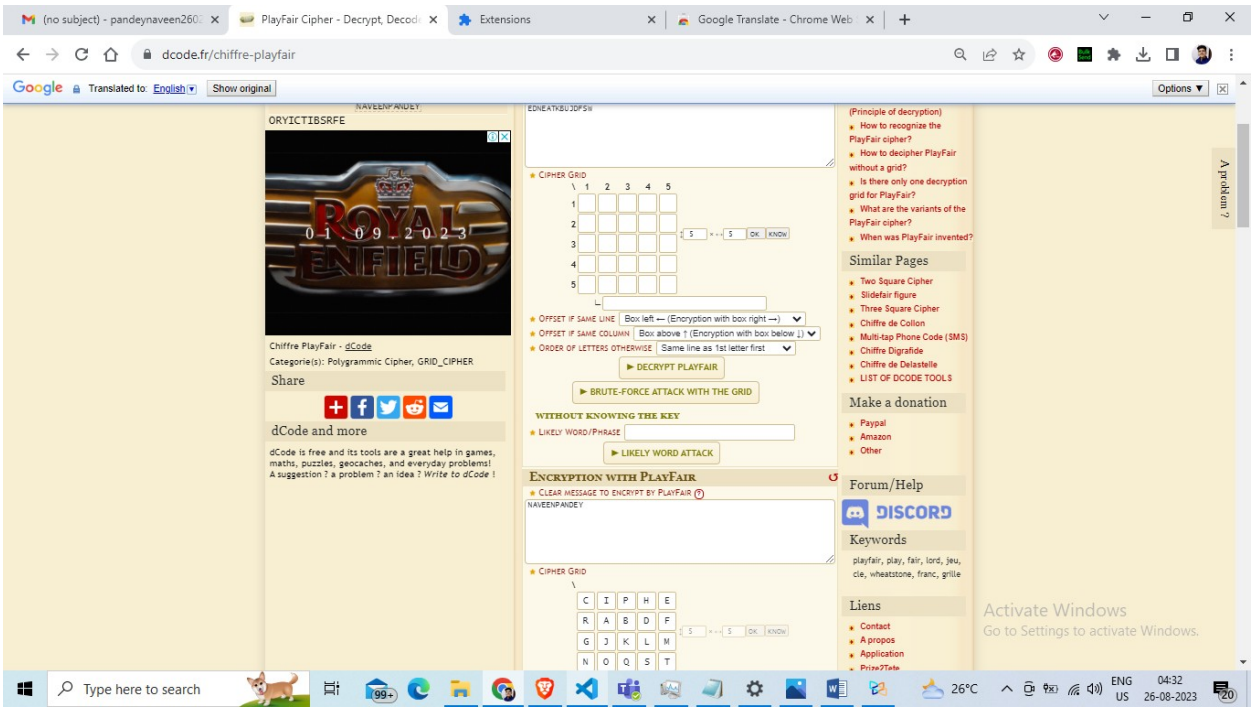
Implementaion:

Playfair Cipher:

Plaintext: NAVEENPANDEY

Cipher grid: CIPHERABDFGJKLMNOQSTUVWXY

Ciphertext: ORYICTIBSRFE



Ciphertext: ODMGROLPRO

Cipher grid: CIPHERABDFGJKLMNOQSTUVWXY

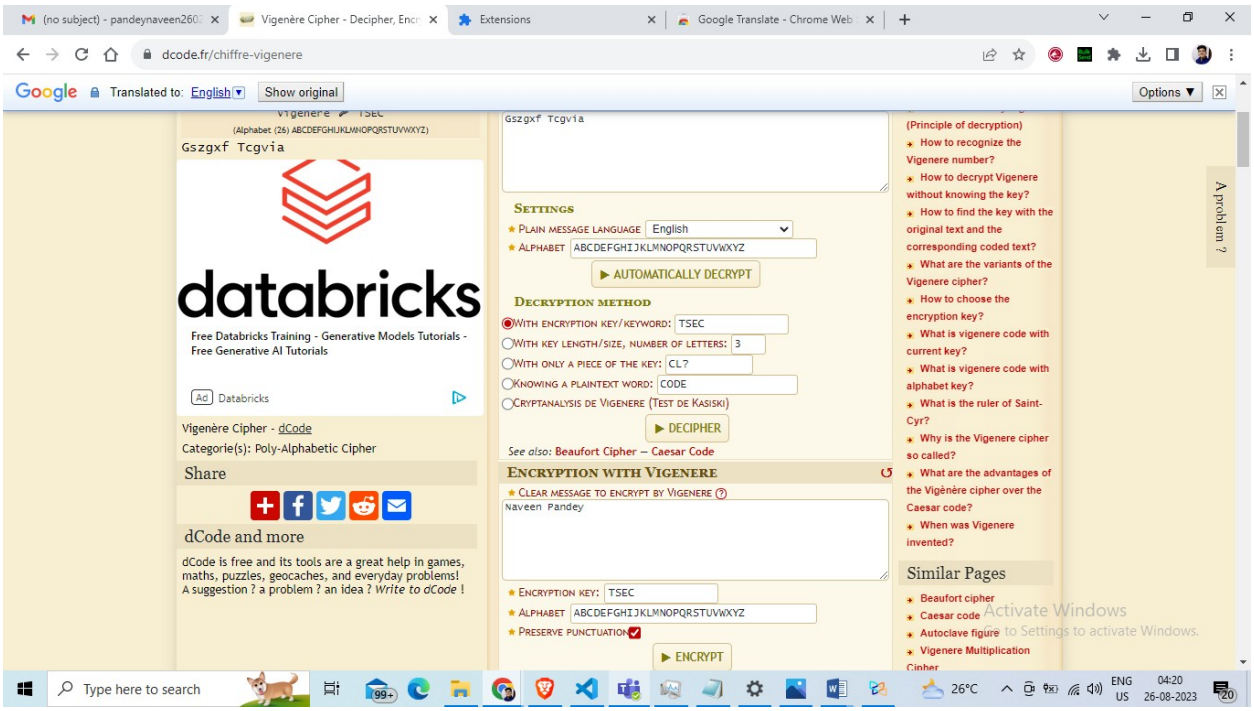
Decrypted text: SALMANKHAN

Vigenere Cipher:

`Input : Plaintext :  Naveen Pandey`

`Keyword :  TSEC`

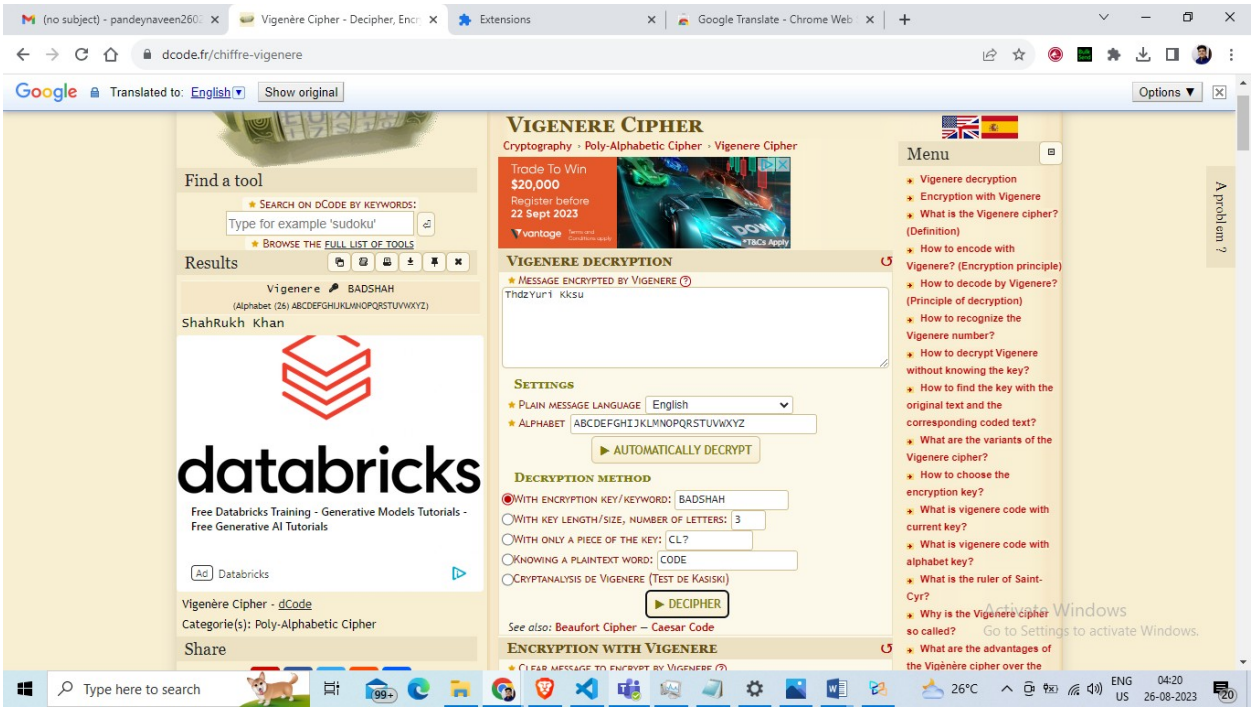
`Output : Encrypted Ciphertext :` Gszgxf Tcgvia



Input : Ciphertext : ThdzYuri Kksu

Keyword : BADSHAH

Output : Decrypted text : ShahRukh Khan



Conclusion:

In conclusion, delving into the implementation of the Playfair and Vigenere ciphers has been an illuminating journey into the realm of classical cryptography. These historical encryption methods have offered valuable insights into the ingenious techniques employed to secure information in earlier times. Through hands-on implementation, I have gained a deeper appreciation for the subtleties of the Playfair matrix and its ability to encrypt digraphs effectively. Similarly, exploring the Vigenere cipher's polyalphabetic approach emphasized the significance of keyword-based encryption and its resistance to frequency analysis. mechanisms.

CNS Assignment 3

Aim: Implementation and analysis of RSA cryptosystem and Digital Signature scheme using RSA

Theory:

RSA (Rivest-Shamir-Adleman):

RSA is a widely used public-key cryptosystem for secure data transmission and digital signatures. It's based on the mathematical properties of large prime numbers. RSA involves a pair of keys: a public key for encryption and a private key for decryption. The security of RSA relies on the difficulty of factoring large semiprime numbers.

Algorithm:

1. Key Generation:

   - Choose two distinct prime numbers, p and q.

   - Calculate n = p * q.

   - Compute the totient φ(n) = (p - 1) * (q - 1).

   - Choose an integer e (usually a small prime, commonly 65537) that is coprime with φ(n).

   - Compute d such that (d * e) % φ(n) = 1.

   - Public key: (e, n)

   - Private key: (d, n)

2. Encryption:

   - Convert the plaintext message into a numeric value m.

   - Compute the ciphertext c = (m^e) % n.

3. Decryption:

   - Compute the plaintext message m = (c^d) % n.

Digital Signature:

A digital signature is a cryptographic technique that provides authenticity, integrity, and non-repudiation for digital messages or documents. It involves using a private key to sign the message and a public key to verify the signature. Digital signatures ensure that the sender of a message is authenticated and that the message has not been tampered with during transmission.

Algorithm:

1. Key Generation:

   - Choose a private key for signing.

   - Compute a corresponding public key for verification.

2. Signing:

   - Hash the message to produce a fixed-length digest.

   - Encrypt the digest using the private key to create the digital signature.

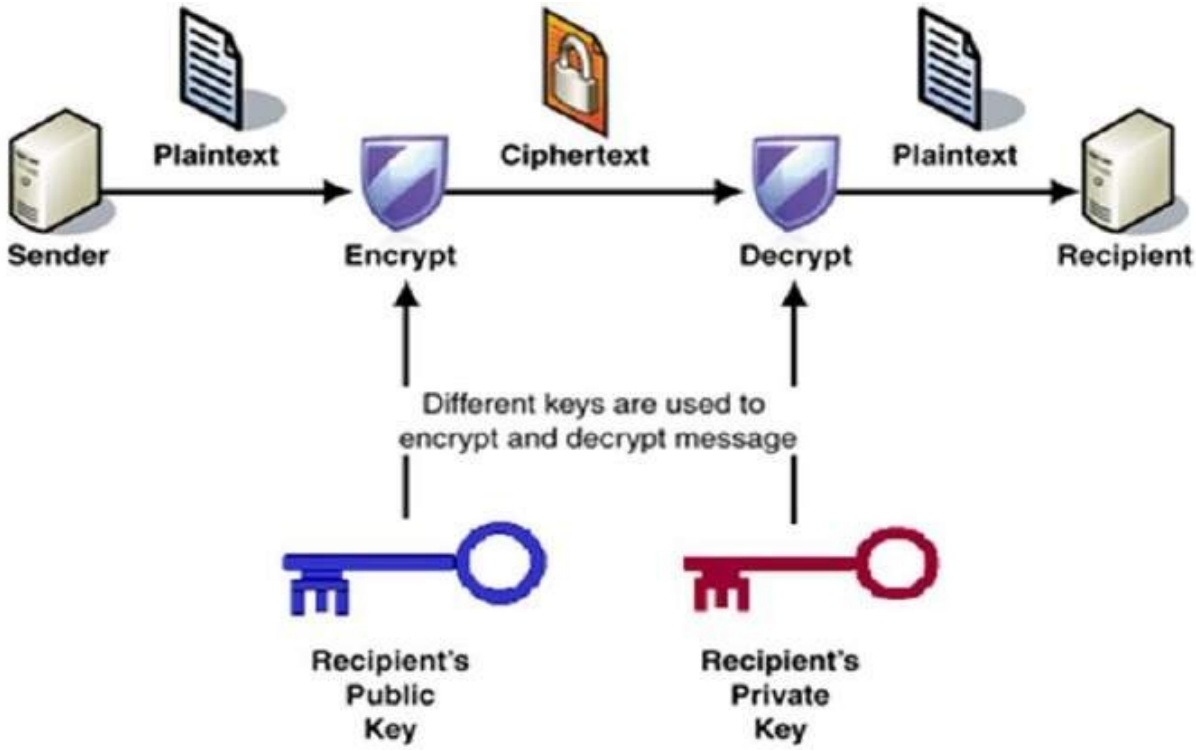
3. Verification:

   - Decrypt the digital signature using the sender's public key to get the digest.

   - Hash the received message to produce a digest.

   - Compare the two digests. If they match, the signature is valid.

Digital signatures are essential for secure communication, online transactions, and authentication of digital documents.



RSA Encryption

Enter Plain Text to Encrypt

Hello I am a Human

Enter Public/Private key

MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAIao
g6+wSDZce/Mqi4o+5Y9r4mvoTdtuEjXZ6RmGr
201ZV2JGlU6b4BQ2kBdY1tLRpZ/kaNSSFedUM9g
9P7RQrUCAwEAAQ==

RSA Key Type: ◉Public key ○Private Key

Select Cipher Type

RSA

Encrypt

Encrypted Output (Base64):

QNDXULc+oOrC9NuLpUuYj60OQ512TBoQtVyHIH
tftpz0ClHnAn+tozTbneWuVmn1FlMXT9unZSwG
RJiAS5YCNw==

RSA Decryption

Enter Encrypted Text to Decrypt (Base64)

QNDXULc+oOrC9NuLpUuYj60OQ512TBoQtVyHI
Htftpz0ClHnAn+tozTbneWuVmn1FlMXT9unZSw
GRJiAS5YCNw==

Enter Public/Private key

WBrCjEBt18fRHRkT+D9nqcwxqBjiJQIgQ5lLiGQ
UzxgM7oxSxvF2kZQ1CzJ9lP+x1fUdq24O7QECI
FJBQGyjvsHcucyC1gvNDE1ApzCMYlkyhuQiBP
Od7C4pAiB3tPw7gPaggcK/u6Y7k4A1ioifO8A
VguKJt+r8DxsAxw==

RSA Key Type: ○Public key ◉Private Key

Select Cipher Type

RSA

Decrypt

Decrypted Output:

Hello I am a Human

Conclusion: Thus we learnt and implemented RSA and digital signature using RSA

CNS Assignment 4

Aim: Study the use of reconnaissance tools like WHOIS, NSLOOKUP, TRACE ROUTE, DIG, NIKTO and DIMITRY to gather information about networks and domain registers

Theory:

WHOIS:

WHOIS is a protocol used to query databases that store registered users or assignees of an Internet resource, such as a domain name, IP address, or Autonomous System Number (ASN). It provides information about the owner, contact details, registration date, and more.

Usage:

WHOIS queries are performed using the WHOIS command or through online WHOIS lookup services. By querying WHOIS databases, you can gather information about domain names, IP addresses, and network allocations.

NSLOOKUP (Name Server Lookup):

NSLOOKUP is a command-line tool used to query DNS (Domain Name System) records. It helps in finding IP addresses associated with domain names and vice versa. It's also used to troubleshoot DNS-related issues.

Usage:

By providing a domain name or IP address as input, NSLOOKUP retrieves DNS records such as A records (IPv4 addresses), AAAA records (IPv6 addresses), MX records (mail servers), and more.

TRACEROUTE (Traceroute or Tracert):

TRACEROUTE is a command-line tool that traces the route taken by packets across a network to a destination host. It displays the IP addresses of routers or hops between the source and destination.

Usage:

TRACEROUTE sends packets with increasing Time-To-Live (TTL) values. Each router decreases the TTL and responds when TTL reaches zero, revealing its IP address. This helps identify network delays or failures.

DIG (Domain Information Groper):

DIG is a command-line tool used to perform DNS queries, similar to NSLOOKUP. It provides information about DNS records, including A, AAAA, MX, CNAME, and more.

Usage:

DIG allows you to specify DNS servers for querying and retrieve detailed DNS records for domain names, helping in DNS troubleshooting and information gathering.

NIKTO:

NIKTO is an open-source web server scanner that identifies vulnerabilities and misconfigurations in web servers and applications. It helps in reconnaissance by discovering potential security issues.

Usage:

NIKTO performs various checks, including outdated software versions, common vulnerabilities, server misconfigurations, and more, against a target web server. It generates a report of findings.

DMITRY (Deep Magic Information Gathering Tool):

Theory:

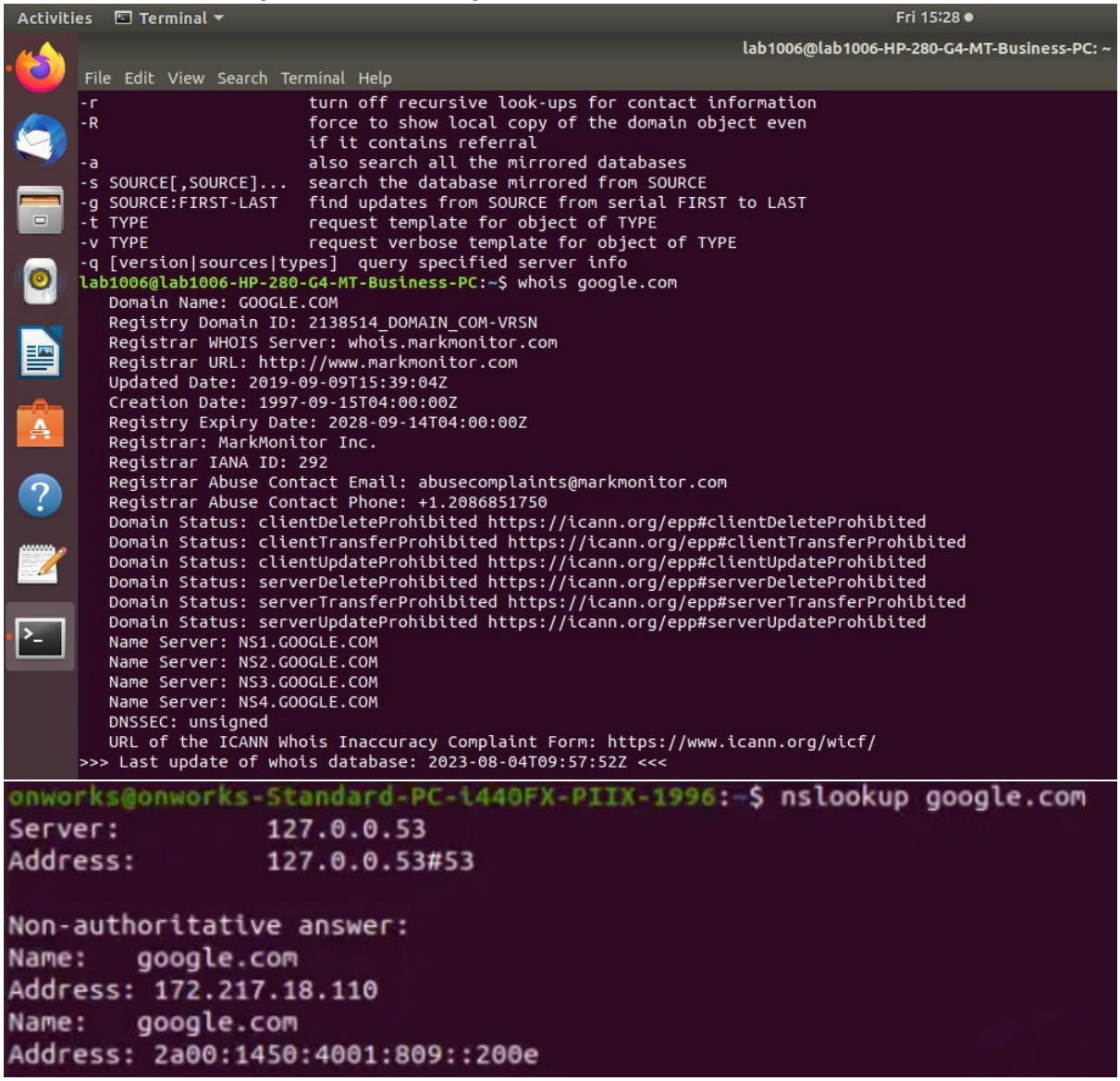
DMITRY is a command-line tool that gathers information about domains, networks, hosts, and people. It's used for reconnaissance to gather public information about a target.

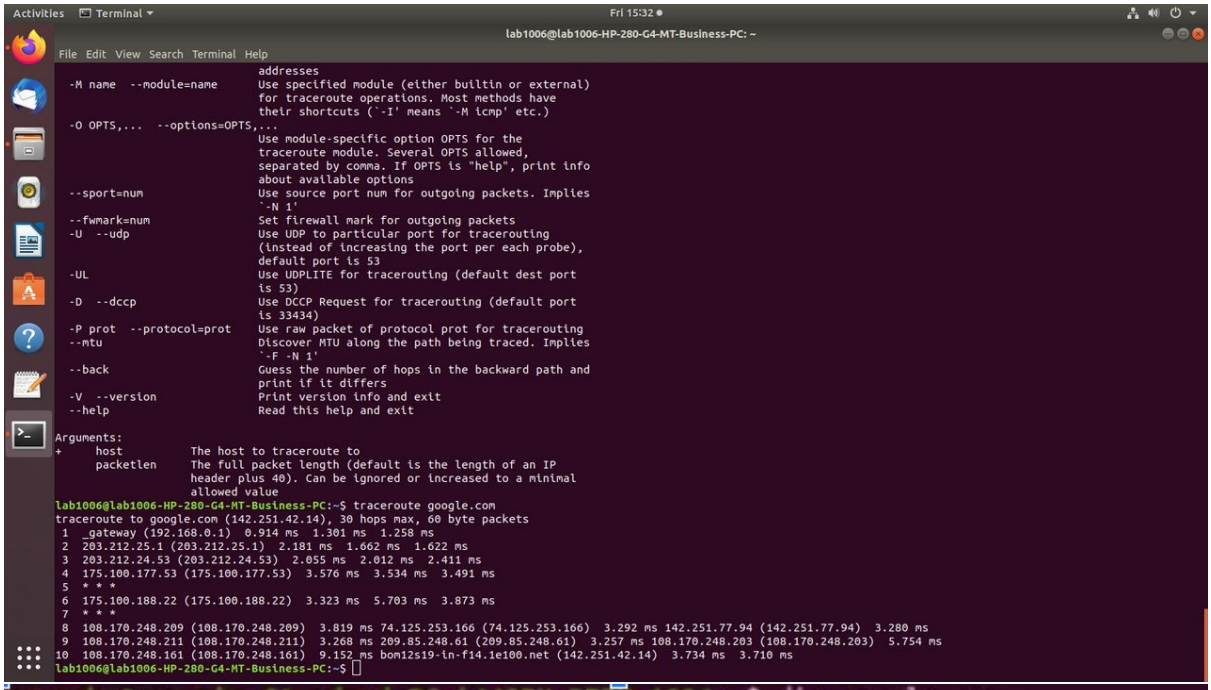
Usage:

DMITRY retrieves data from various sources, including WHOIS databases, DNS records, search engines, and more. It compiles a report containing information like domain ownership, network information, and related hosts.

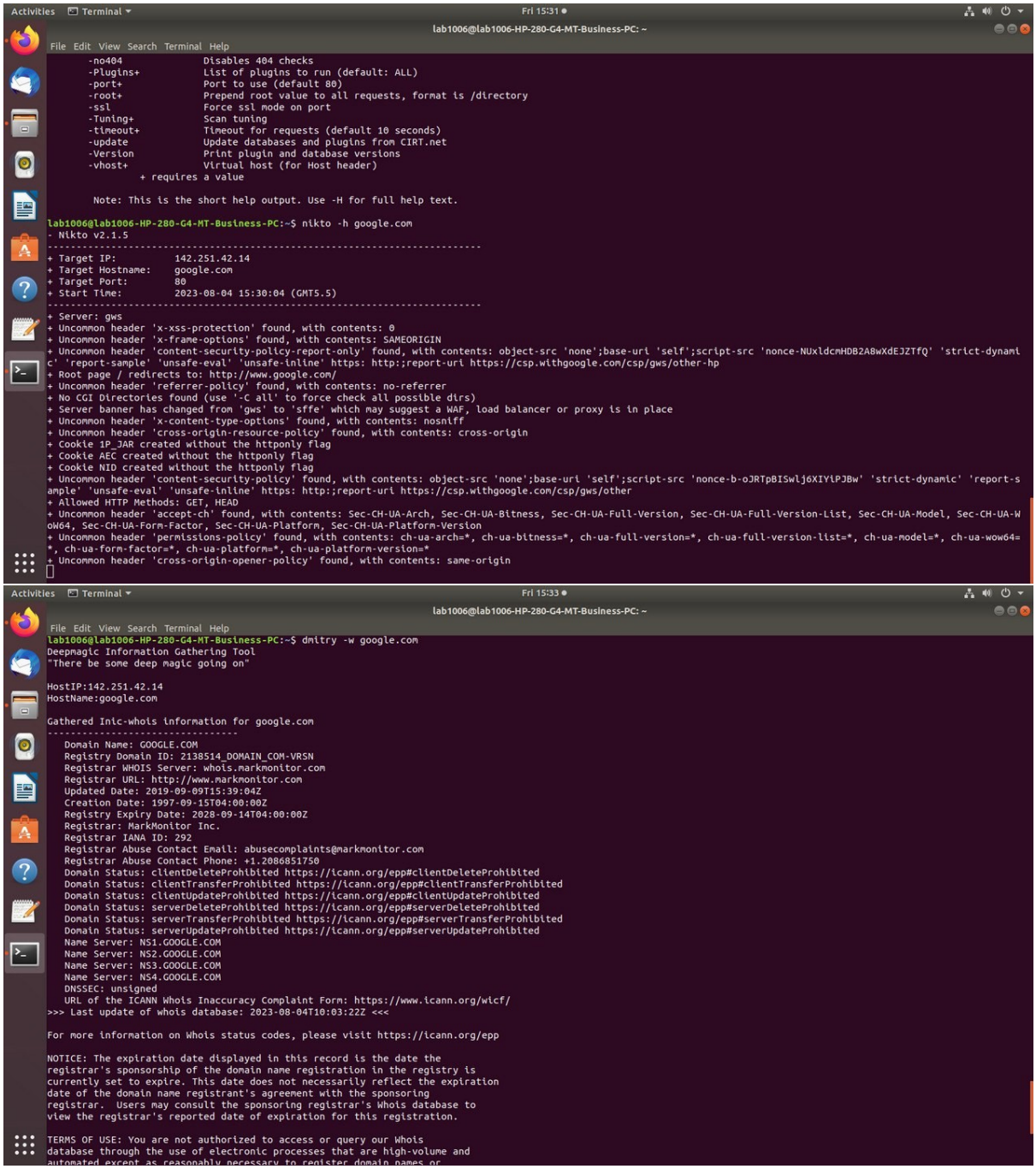
These tools are widely used in ethical hacking, penetration testing, and security assessment to gather information about target systems and networks. Always ensure you have proper authorization before using

these tools on any network or system.






```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ dig google.com

; <<>> DiG 9.16.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60699
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             199     IN      A       142.250.185.142

;; Query time: 8 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fr Aug 04 11:26:06 CEST 2023
;; MSG SIZE  rcvd: 55
```



Conclusion: Thus we implmented various reconnaissance tools to gather information about domain registers and networks

Lab Assignment 5

Aim: To explore Hashdeep tool in kali linux for generating, matching andauditing hash of files.

Lab Outcome Attainment: LO2

Theory:

Hashdeep is a command-line utility for computing and verifying hash values (checksums) of files and directories. It is a versatile and powerful tool primarily used for data integrity verification and digital forensics. Hashdeep can calculate multiple hash values (e.g., MD5, SHA-1, SHA-256, SHA-512) for files and directories and store them in hash databases. You can then use these hash databases to verify the integrity of your files at a later time by comparing the computed hash values with the stored ones. Some key features and use cases of Hashdeep include:

1. **Data Integrity Verification**: Hashdeep is commonly used to ensure that files have not been tampered with or corrupted over time. By periodically recalculating hash values and comparing them to the stored values, you can detect any unauthorized changes.

2. **Forensics and Investigations**: Digital forensics experts use Hashdeep to create hash databases of evidence and verify its integrity during investigations. This helps ensure that the data remains unchanged throughout the legal process.

3. **Comparing Directories**: You can use Hashdeep to compare two directories to find differences between them, even if the file names have changed. This is useful for backup verification and synchronization tasks.

4. **Recursive Hashing**: Hashdeep can recursively calculate hash values for directories and subdirectories, making it efficient for processing large and complex directory structures.

5. **Cross-Platform**: Hashdeep is available for various operating systems, including Linux, macOS, and Windows, making it a versatile tool for cross-platform use.
6. **Support for Multiple Hash Algorithms**: It supports multiple hash algorithms, including MD5, SHA-1, SHA-256, SHA-512, and others, allowing you to choose the level of security and performance you need.

How to use **hashdeep :**

1. To check the version of Hashdeep - *Hashdeep -V*

2. To display help about Hashdeep - *Hashdeep -h* or *Hashdeep -hh*

3. To display the manual page of Hashdeep- *man Hashdeep*

4. To display the manual page of any specific hash algorithm supported by Hashdeep- *man md5deep*

5. To hash a file - *Hashdeep filename*

6. To supress any error messages- *Hashdeep -s filename*

7. To apply multiple hash algorithms than default-

   Hashdeep -c md5,sha1,sha256,tiger filename

8. To hash multiple files (say all text files) using md5

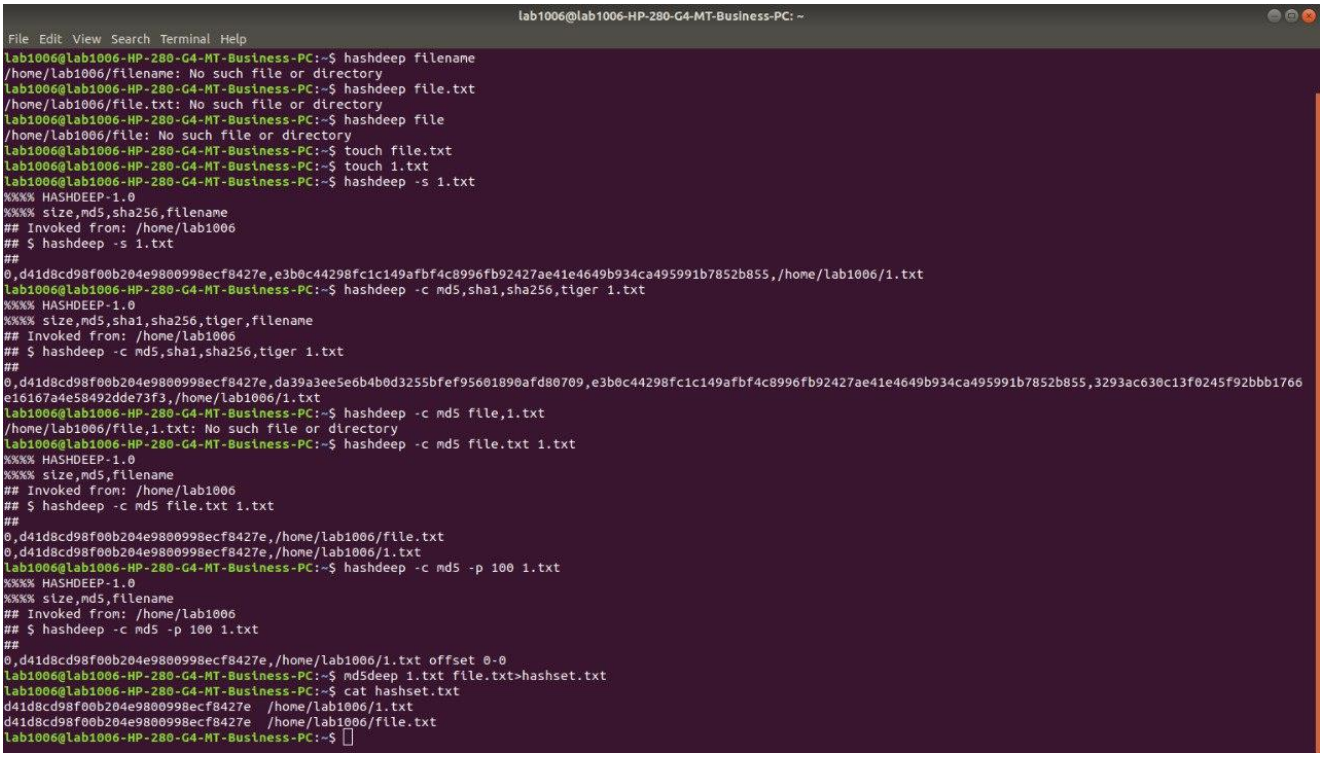
   *Hashdeep -c md5 *.txt*

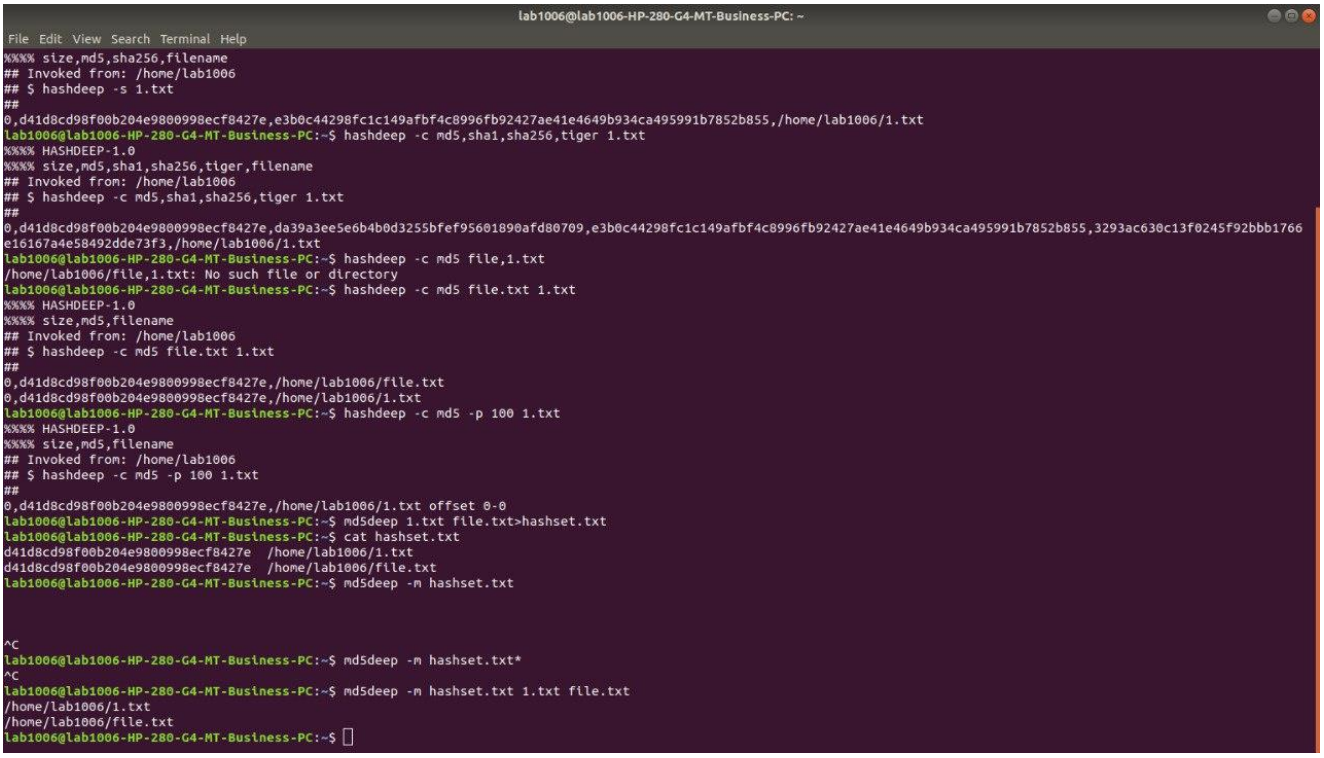
9. To hash multiple files (say all text files) using md5 and sha1

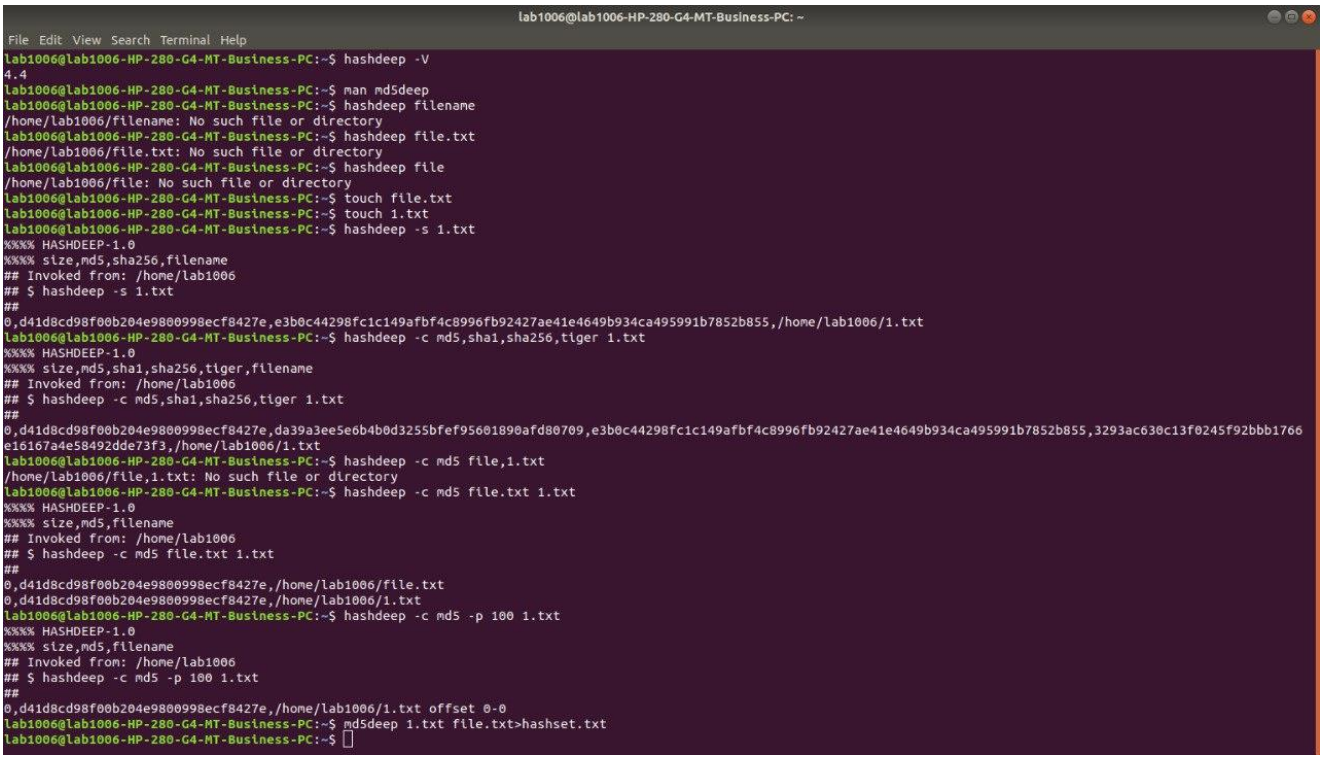
   *Hashdeep -c md5,sha1 *.txt*

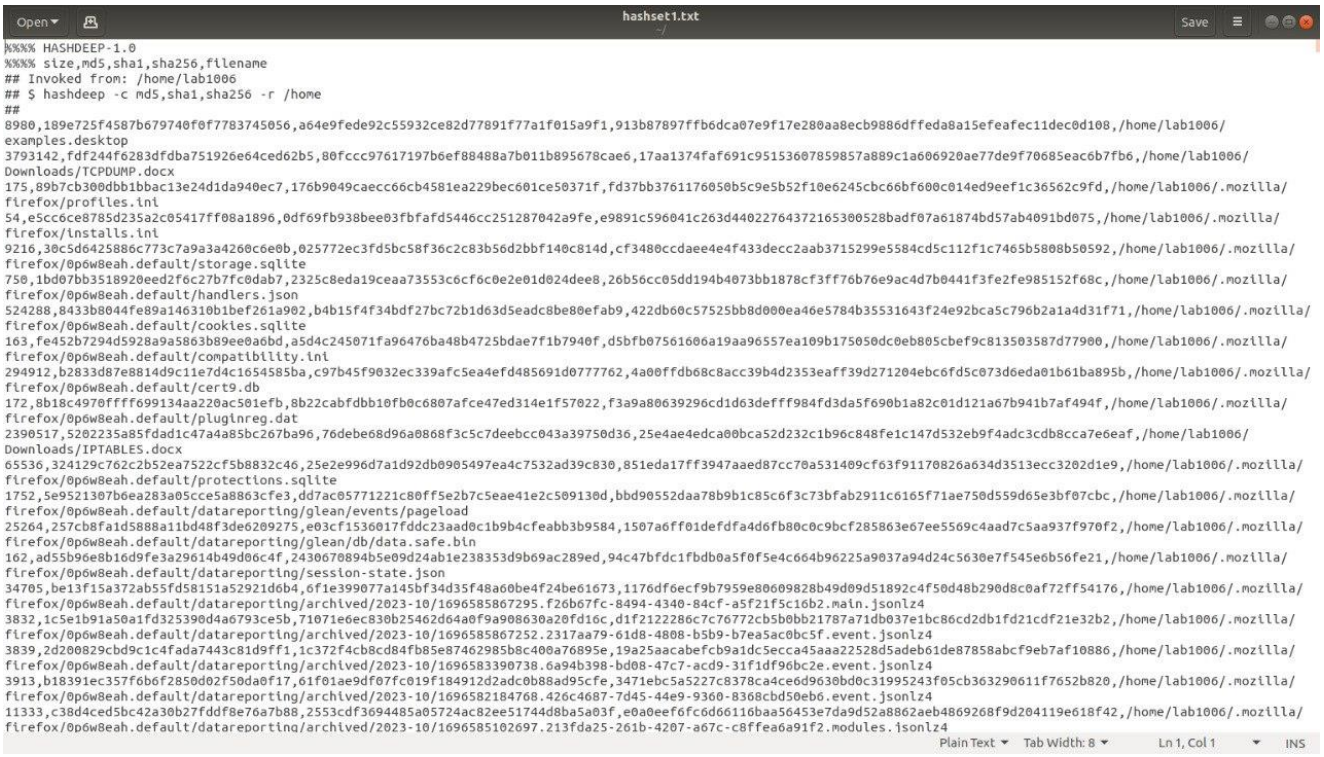
10. Hashing block of files-
    Hashdeep -c md5 -p 100 example.txt

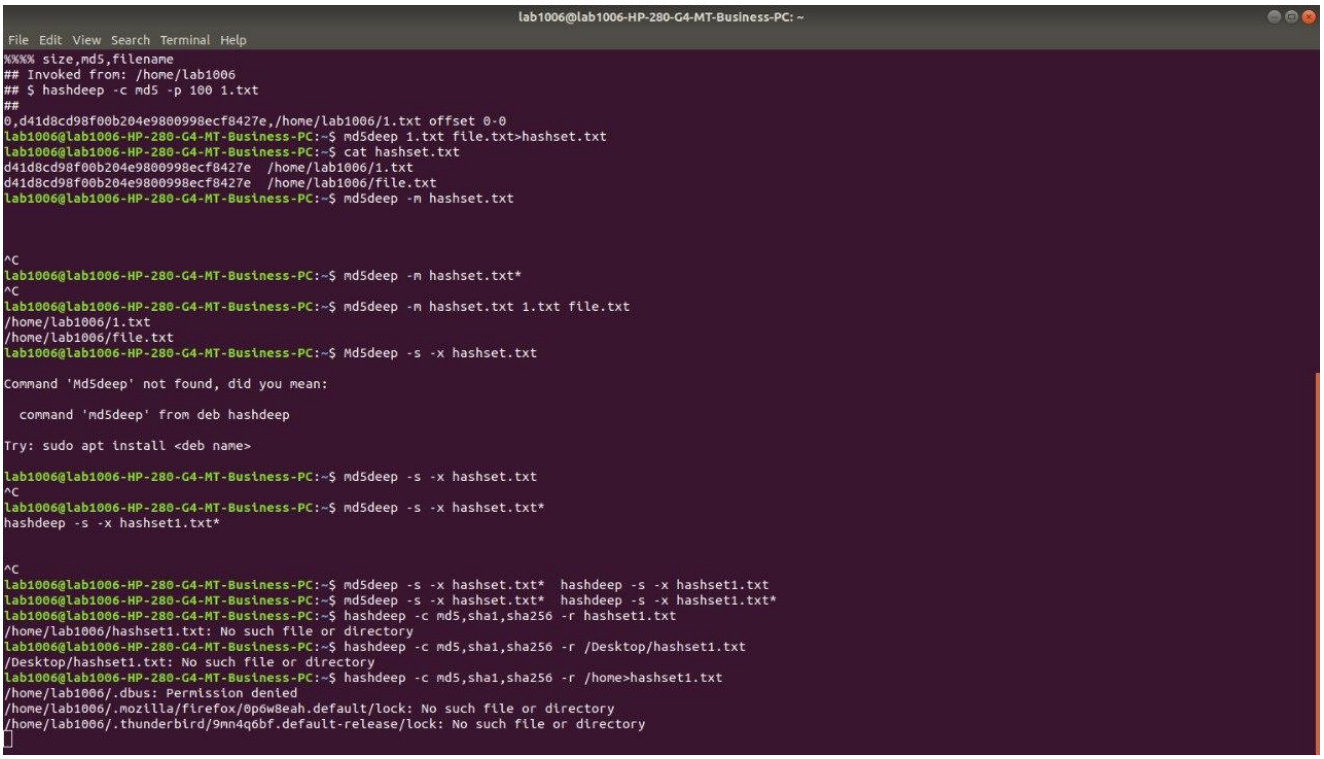
Output :


lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep filename
/home/lab1006/filename: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file.txt
/home/lab1006/file.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file
/home/lab1006/file: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch 1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
Invoked from: /home/lab1006
$ hashdeep -s 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,sha256,tiger,filename
Invoked from: /home/lab1006
$ hashdeep -c md5,sha1,sha256,tiger 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,da39a3ee5e6b4b0d3255bfef95601890afd80709,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb1766
e16167a4e58492dde73f3,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt
/home/lab1006/file,1.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 file.txt 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file.txt
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 -p 100 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset.txt
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/1.txt
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$


lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
%%%% size,md5,sha256,filename
Invoked from: /home/lab1006
$ hashdeep -s 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,sha256,tiger,filename
Invoked from: /home/lab1006
$ hashdeep -c md5,sha1,sha256,tiger 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,da39a3ee5e6b4b0d3255bfef95601890afd80709,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb1766
e16167a4e58492dde73f3,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt
/home/lab1006/file,1.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 file.txt 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file.txt
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 -p 100 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset.txt
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/1.txt
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt
^C
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt*
^C
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt 1.txt file.txt
/home/lab1006/1.txt
/home/lab1006/file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$


lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -V
4.4
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man md5deep
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep filename
/home/lab1006/filename: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file.txt
/home/lab1006/file.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file
/home/lab1006/file: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch 1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
Invoked from: /home/lab1006
$ hashdeep -s 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,sha256,tiger,filename
Invoked from: /home/lab1006
$ hashdeep -c md5,sha1,sha256,tiger 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,da39a3ee5e6b4b0d3255bfef95601890afd80709,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb1766
e16167a4e58492dde73f3,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt
/home/lab1006/file,1.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 file.txt 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file.txt
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
Invoked from: /home/lab1006
$ hashdeep -c md5 -p 100 1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$


Open
hashset1.txt
Save
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,sha256,filename
Invoked from: /home/lab1006
$ hashdeep -c md5,sha1,sha256 -r /home
##
8980,189e725f4587b679740f0f7783745056,a64e9fede92c55932ce82d77891f77a1f015a9f1,913b87897ffb6dca07e9f17e280aa8ecb9886dffeda8a15efeafec11dec0d108,/home/lab1006/
examples.desktop
3793142,fdf244f6283dfdba751926e64ced62b5,80fccc97617197b6ef88488a7b011b895678cae6,17aa1374faf691c95153607859857a889c1a606920ae77de9f70685eac6b7fb6,/home/lab1006/
Downloads/TCPDUMP.docx
175,89b7cb300dbb1bbac13e24d1da940ec7,176b9049caecc66cb4581ea229bec601ce50371f,fd37bb3761176050b5c9e5b52f10e6245cbc66bf600c014ed9eef1c36562c9fd,/home/lab1006/.mozilla/
firefox/profiles.ini
54,e5cc6ce8785d235a2c05417ff08a1896,0df69fb938bee03fbfafd5446cc251287042a9fe,e9891c596041c263d4402276437216530052 8badf07a61874bd57ab4091bd075,/home/lab1006/.mozilla/
firefox/installs.ini
9216,30c5d6425886c773c7a9a3a4260c6e0b,025772ec3fd5bc58f36c2c83b56d2bbf140c814d,cf3480ccdaee4e4f433decc2aab3715299e5584cd5c112f1c7465b5808b50592,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/storage.sqlite
750,1bd07bb3518920eed2f6c27b7fc0dab7,2325c8eda19ceaa73553c6cf6c0e2e01d024dee8,26b56cc05dd194b4073bb1878cf3ff76b76e9ac4d7b0441f3fe2fe985152f68c,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/handlers.json
524288,8433b8044fe89a146310b1bef261a902,b4b15f4f34bdf27bc72b1d63d5eadc8be80efab9,422db60c57525bb8d000ea46e5784b35531643f24e92bca5c796b2a1a4d31f71,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/cookies.sqlite
163,fe452b7294d5928a9a5863b89ee0a6bd,a5d4c245071fa96476ba48b4725bdae7f1b7940f,d5bfb07561606a19aa96557ea109b175050dc0eb805cbef9c813503587d77900,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/compatibility.ini
294912,b2833d87e8814d9c11e7d4c1654585ba,c97b45f9032ec339afc5ea4efd485691d0777762,4a00ffdb68c8acc39b4d2353eaff39d271204ebc6fd5c073d6eda01b61ba895b,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/cert9.db
172,8b18c4970ffff699134aa220ac501efb,8b22cabfdbb10fb0c6807afce47ed314e1f57022,f3a9a80639296cd1d63deff984fd3da5f690b1a82c01d121a67b941b7af494f,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/pluginreg.dat
2390517,5202235a85fdad1c47a4a85bc267ba96,76debe68d96a0868f3c5c7deebcc043a39750d36,25e4ae4edca00bca52d232c1b96c848fe1c147d532eb9f4adc3cdb8cca7e6eaf,/home/lab1006/
Downloads/IPTABLES.docx
65536,324129c762c2b52ea7522cf5b8832c46,25e2e996d7a1d92db0905497ea4c7532ad39c830,851eda17ff3947aaed87cc70a531409cf63f91170826a634d3513ecc3202d1e9,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/protections.sqlite
1752,5e9521307b6ea283a05cce5a8863cfe3,dd7ac05771221c80ff5e2b7c5eae41e2c509130d,bbd90552daa78b9b1c85c6f3c73bfab2911c6165f71ae750d559d65e3bf07cbc,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/glean/events/pageload
25264,257cb8fa1d5888a11bd48f3de6209275,e03cf1536017fddc23aad0c1b9b4cfeabb3b9584,1507a6ff01defdfa4d6fb80c0c9bcf285863e67ee5569c4aad7c5aa937f970f2,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/glean/db/data.safe.bin
162,ad55b96e8b16d9fe3a29614b49d06c4f,2430670894b5e09d24ab1e238353d9b69ac289ed,94c47bfdc1fbdb0a5f0f5e4c664b96225a9037a94d24c5630e7f545e6b56fe21,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/session-state.json
34705,be13f15a372ab55fd58151a52921d6b4,6f1e399077a145bf34d35f48a60be4f24be61673,1176df6ecf9b7959e80609828b49d09d51892c4f50d48b290d8c0af72ff54176,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/archived/2023-10/1696585867295.f26b67fc-8494-4340-84cf-a5f21f5c16b2.main.jsonlz4
3832,1c5e1b91a50a1fd325390d4a6793ce5b,71071e6ec830b25462d64a0f9a908630a20fd16c,d1f2122286c7c76772cb5b0bb21787a71db037e1bc86cd2db1fd21cdf21e32b2,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/archived/2023-10/1696585867252.2317aa79-61d8-4808-b5b9-b7ea5ac0bc5f.event.jsonlz4
3839,2d200829cbd9c1c4fada7443c81d9ff1,1c372f4cb8cd84fb85e87462985b8c400a76895e,19a25aacabefcb9a1dc5ecca45aaa22528d5adeb61de87858abcf9eb7af10886,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/archived/2023-10/1696583390738.6a94b398-bd08-47c7-acd9-31f1df96bc2e.event.jsonlz4
3913,b18391ec357f6b6f2850d02f50da0f17,61f01ae9df07fc019f184912d2adc0b88ad95cfe,3471ebc5a5227c8378ca4ce6d9630bd0c31995243f05cb363290611f7652b820,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/archived/2023-10/1696582184768.426c4687-7d45-44e9-9360-8368cbd50eb6.event.jsonlz4
11333,c38d4ced5bc42a30b27fddf8e76a7b88,2553cdf3694485a05724ac82ee51744d8ba5a03f,e0a0eef6fc6d66116baa56453e7da9d52a8862aeb4869268f9d204119e618f42,/home/lab1006/.mozilla/
firefox/0p6w8eah.default/datareporting/archived/2023-10/1696585102697.213fda25-261b-4207-a67c-c8ffea6a91f2.modules.jsonlz4
Plain Text
Tab Width: 8
Ln 1, Col 1
INS



Conclusion :

We understood hashdeep and its versatile command-line utility that computes and verifies checksums (hash values) for files and directories, offering data integrity assurance and digital forensics capabilities. It supports multiple hash algorithms, making it a reliable tool for detecting file tampering and ensuring the integrity of data across different platforms.

Lab Assignment 7

AIM: Study of packet sniffer tools TCPDUMP.

LO3: Explore the different network reconnaissance tools to gather information about networks.

THEORY:

What is TCPDUMP and how to install it?

Tcpdump is a command-line packet analyzer that allows you to capture and analyze network traffic in real-time. It's commonly used for troubleshooting network issues, analyzing network behavior, and diagnosing problems related to network communication. tcpdump captures packets as they travel through a network interface and provides detailed information about each packet, including source and destination addresses, protocol information, payload data, and more.

Linux (Debian/Ubuntu):

Open a terminal and run the following command to install tcpdump: sudo apt-get update sudo apt-get install tcpdump

Explain various commands in tcpdump to capture different types of packets.

tcpdump provides a wide range of commands and options to capture and analyze different types of packets. Here are some common tcpdump commands and filters to capture specific types of packets:

1. Capture All Traffic on a Specific Interface:

sudo tcpdump -i eth0

This captures all traffic on the "eth0" network interface.

2. Capture Traffic to or from a Specific IP Address:

sudo tcpdump host 192.168.1.100

This captures all traffic to or from the IP address "192.168.1.100".

3. Capture Traffic on a Specific Port:

sudo tcpdump port 80

This captures all traffic on port 80.

4. Capture Traffic Using a Specific Protocol:

sudo tcpdump icmp

This captures ICMP (ping) traffic.

5. Capture Traffic from a Specific Source IP:

sudo tcpdump src 192.168.1.200

This captures traffic originating from IP address "192.168.1.200".

6. Capture Traffic to a Specific Destination IP:

sudo tcpdump dst 192.168.1.100

This captures traffic directed to IP address "192.168.1.100".

7. Capture Traffic on a Specific Port Using a Protocol:

sudo tcpdump udp port 53

This captures UDP traffic on port 53 (DNS).

8. Capture Traffic Using a Combination of Filters:

sudo tcpdump src 192.168.1.100 and port 22

This captures traffic originating from IP address "192.168.1.100" and using port 22 (SSH).

9. Capture Traffic with Specific Packet Size:

sudo tcpdump greater 1000

This captures packets larger than 1000 bytes.

10. Capture Specific Number of Packets:

sudo tcpdump -c 10

This captures 10 packets and then exits.

11. Capture Packets Using Hexadecimal Filter:

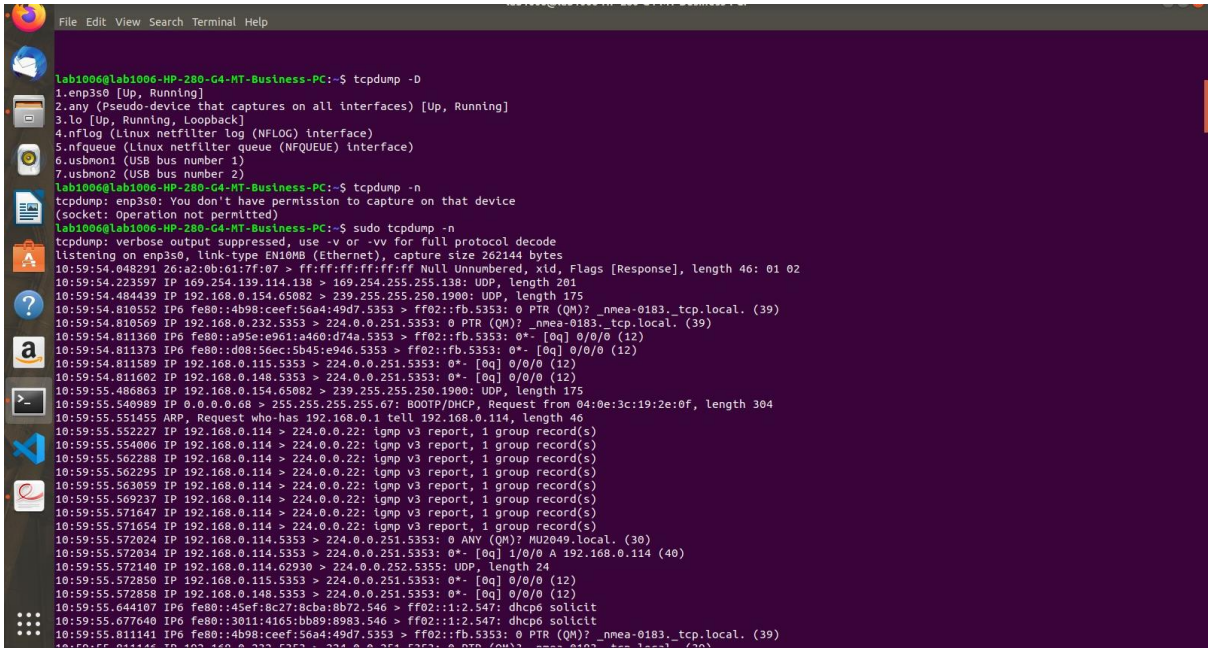
sudo tcpdump -X 'tcp[13] & 2 != 0'

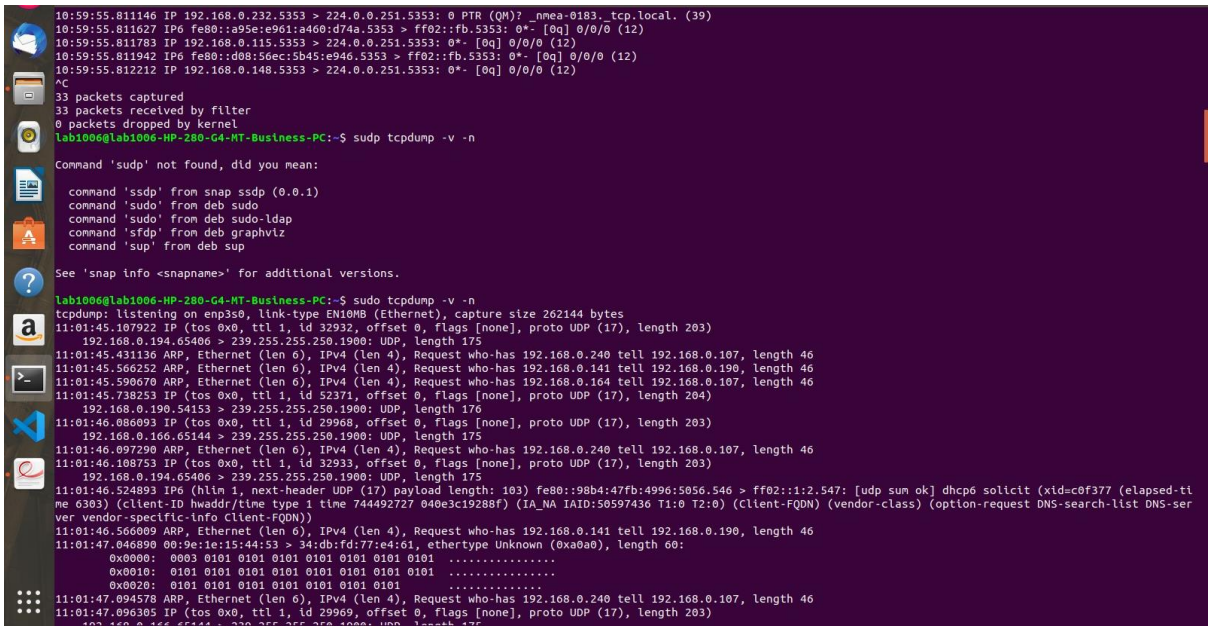
This captures only SYN packets (TCP packets with the SYN flag set).

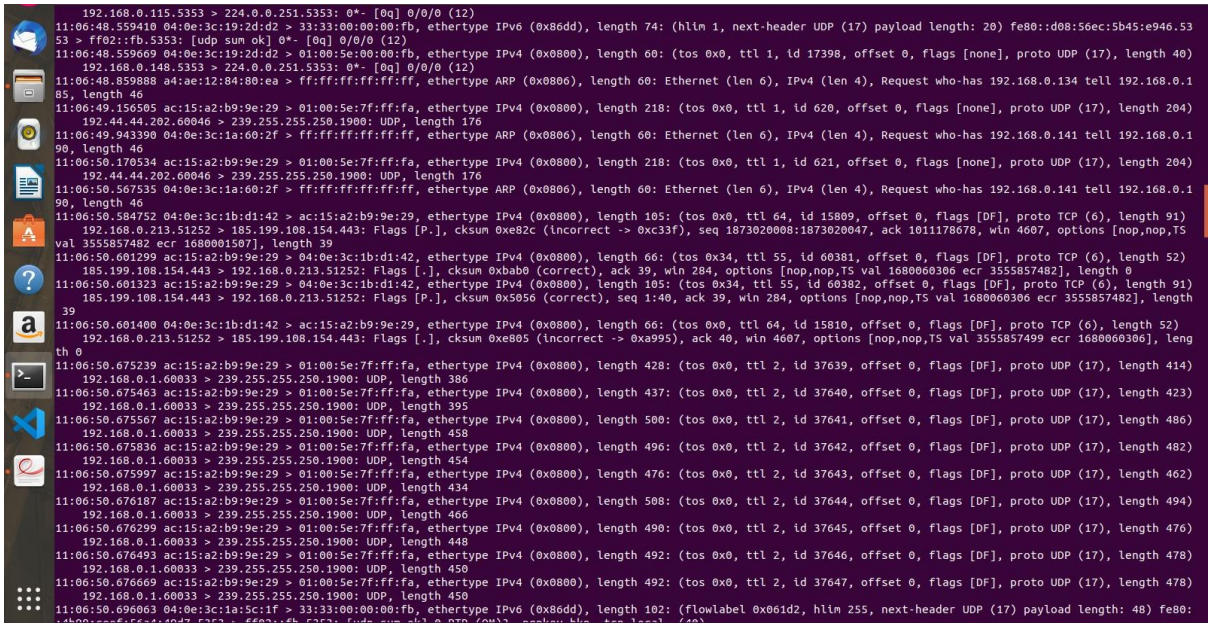
12. Capture and Save Output to a File: sudo tcpdump -i eth0 -w output.pcap

This captures traffic on the "eth0" interface and saves it to the "output.pcap" file.

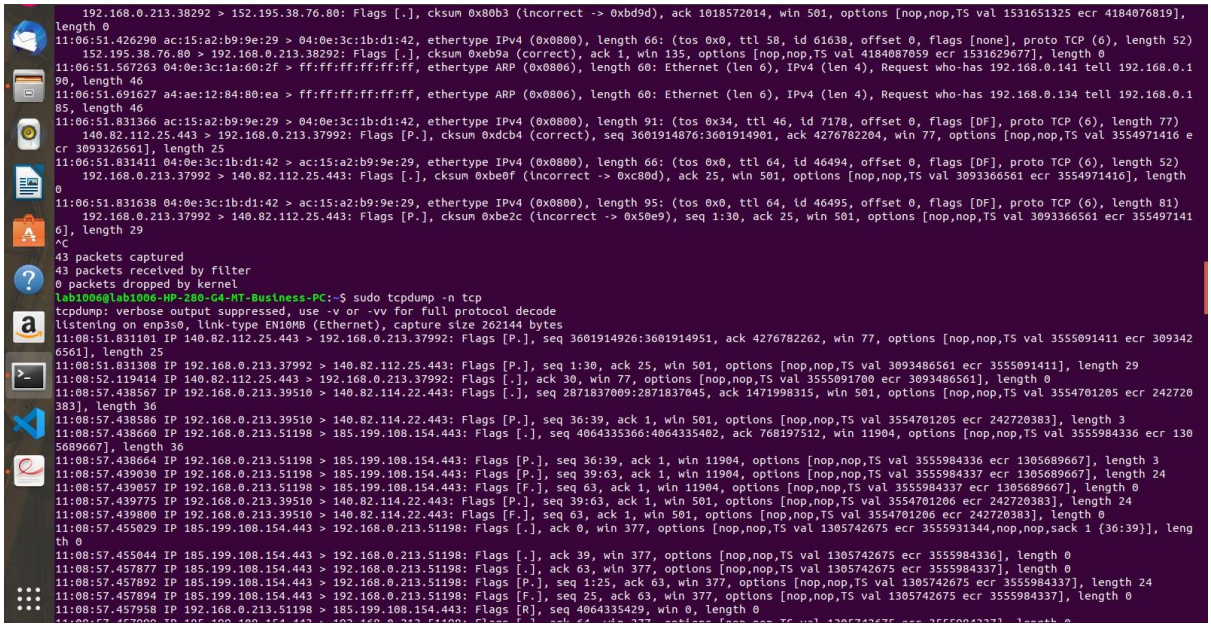
OUTPUT


File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tcpdump -D
1.enp3s0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.nflog (Linux netfilter log (NFLOG) interface)
5.nfqueue (Linux netfilter queue (NFQUEUE) interface)
6.usbmon1 (USB bus number 1)
7.usbmon2 (USB bus number 2)
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tcpdump -n
tcpdump: enp3s0: You don't have permission to capture on that device
(socket: Operation not permitted)
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:59:54.048291 26:a2:0b:61:7f:07 > ff:ff:ff:ff:ff:ff Null Unnumbered, xid, Flags [Response], length 46: 01 02
10:59:54.223597 IP 169.254.139.114.138 > 169.254.255.255.138: UDP, length 201
10:59:54.484439 IP 192.168.0.154.65082 > 239.255.255.250.1900: UDP, length 175
10:59:54.810552 IP6 fe80::4b98:ceef:56a4:49d7.5353 > ff02::fb.5353: 0 PTR (QM)? _nmea-0183._tcp.local. (39)
10:59:54.810569 IP 192.168.0.232.5353 > 224.0.0.251.5353: 0 PTR (QM)? _nmea-0183._tcp.local. (39)
10:59:54.811360 IP6 fe80::a95e:e961:a460:d74a.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
10:59:54.811373 IP6 fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
10:59:54.811589 IP 192.168.0.115.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
10:59:54.811602 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
10:59:55.486863 IP 192.168.0.154.65082 > 239.255.255.250.1900: UDP, length 175
10:59:55.540989 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:19:2e:0f, length 304
10:59:55.551455 ARP, Request who-has 192.168.0.1 tell 192.168.0.114, length 46
10:59:55.552227 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.554006 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.562288 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.562295 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.563059 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.569237 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.571647 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.571654 IP 192.168.0.114 > 224.0.0.22: igmp v3 report, 1 group record(s)
10:59:55.572024 IP 192.168.0.114.5353 > 224.0.0.251.5353: 0 ANY (QM)? MU2049.local. (30)
10:59:55.572034 IP 192.168.0.114.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 A 192.168.0.114 (40)
10:59:55.572140 IP 192.168.0.114.62930 > 224.0.0.252.5355: UDP, length 24
10:59:55.572850 IP 192.168.0.115.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
10:59:55.572858 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
10:59:55.644107 IP6 fe80::45ef:8c27:8cba:8b72.546 > ff02::1:2.547: dhcp6 solicit
10:59:55.677640 IP6 fe80::3011:4165:bb89:8983.546 > ff02::1:2.547: dhcp6 solicit
10:59:55.811141 IP6 fe80::4b98:ceef:56a4:49d7.5353 > ff02::fb.5353: 0 PTR (QM)? _nmea-0183._tcp.local. (39)

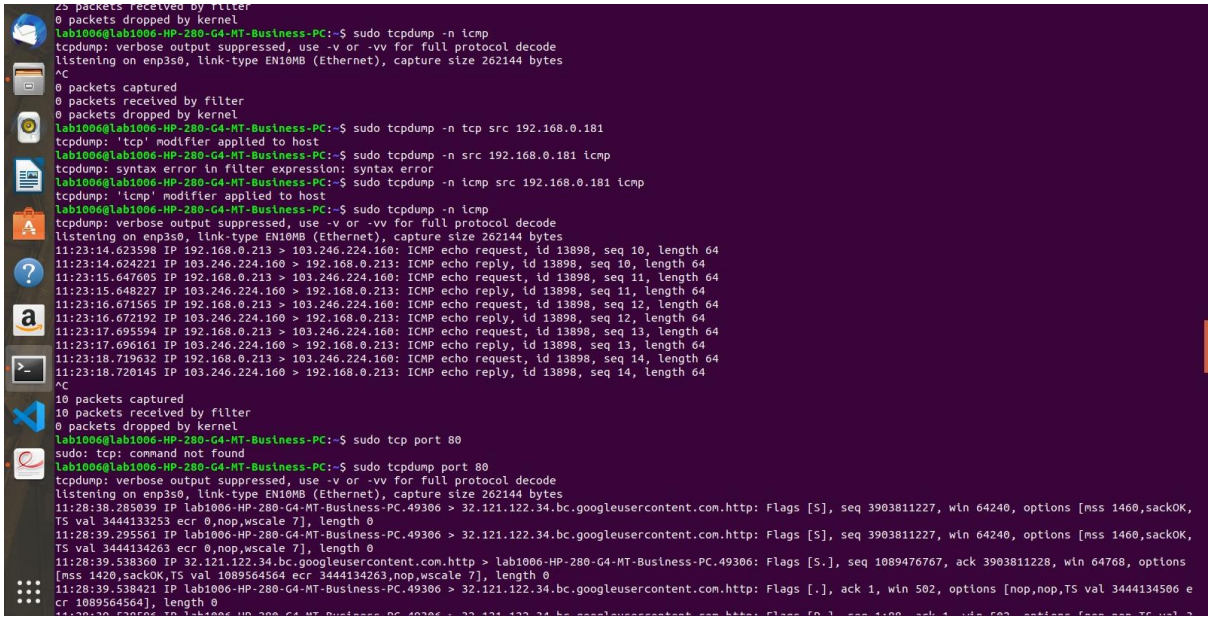

10:59:55.811146 IP 192.168.0.232.5353 > 224.0.0.251.5353: 0 PTR (QM)? _nmea-0183._tcp.local. (39)
10:59:55.811627 IP6 fe80::a95e:e961:a460:d74a.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
10:59:55.811783 IP 192.168.0.115.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
10:59:55.811942 IP6 fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
10:59:55.812212 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
^C
33 packets captured
33 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudp tcpdump -v -n
Command 'sudp' not found, did you mean:
command 'ssdp' from snap ssdp (0.0.1)
command 'sudo' from deb sudo
command 'sudo' from deb sudo-ldap
command 'sfdp' from deb graphviz
command 'sup' from deb sup
See 'snap info <snapname>' for additional versions.
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -v -n
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:01:45.107922 IP (tos 0x0, ttl 1, id 32932, offset 0, flags [none], proto UDP (17), length 203)
192.168.0.194.65406 > 239.255.255.250.1900: UDP, length 175
11:01:45.431136 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.240 tell 192.168.0.107, length 46
11:01:45.566252 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.141 tell 192.168.0.190, length 46
11:01:45.590670 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.164 tell 192.168.0.107, length 46
11:01:45.738253 IP (tos 0x0, ttl 1, id 52371, offset 0, flags [none], proto UDP (17), length 204)
192.168.0.190.54153 > 239.255.255.250.1900: UDP, length 176
11:01:46.086093 IP (tos 0x0, ttl 1, id 29968, offset 0, flags [none], proto UDP (17), length 203)
192.168.0.166.65144 > 239.255.255.250.1900: UDP, length 175
11:01:46.097290 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.240 tell 192.168.0.107, length 46
11:01:46.108753 IP (tos 0x0, ttl 1, id 32933, offset 0, flags [none], proto UDP (17), length 203)
192.168.0.194.65406 > 239.255.255.250.1900: UDP, length 175
11:01:46.524893 IP6 (hlim 1, next-header UDP (17) payload length: 103) fe80::98b4:47fb:4996:5056.546 > ff02::1:2.547: [udp sum ok] dhcp6 solicit (xid=c0f377 (elapsed-time 6303) (client-ID hwaddr/time type 1 time 744492727 040e3c19288f) (IA_NA IAID:50597436 T1:0 T2:0) (client-FQDN) (vendor-class) (option-request DNS-search-list DNS-server vendor-specific-info Client-FQDN))
11:01:46.566009 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.141 tell 192.168.0.190, length 46
11:01:47.046890 00:9e:1e:15:44:53 > 34:db:fd:77:e4:61, ethertype Unknown (0xa0a0), length 60:
0x0000: 0003 0101 0101 0101 0101 0101 0101 0101 ....
0x0010: 0101 0101 0101 0101 0101 0101 0101 0101 ....
0x0020: 0101 0101 0101 0101 0101 0101 0101 0101 ....
11:01:47.094578 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.240 tell 192.168.0.107, length 46
11:01:47.096305 IP (tos 0x0, ttl 1, id 29969, offset 0, flags [none], proto UDP (17), length 203)

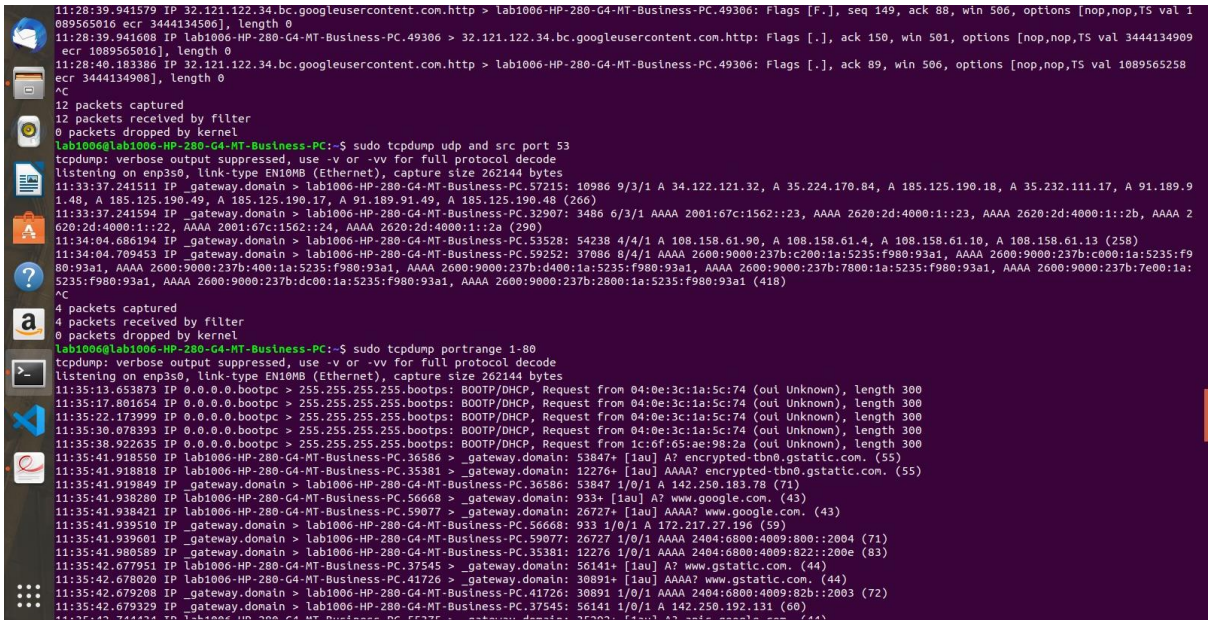


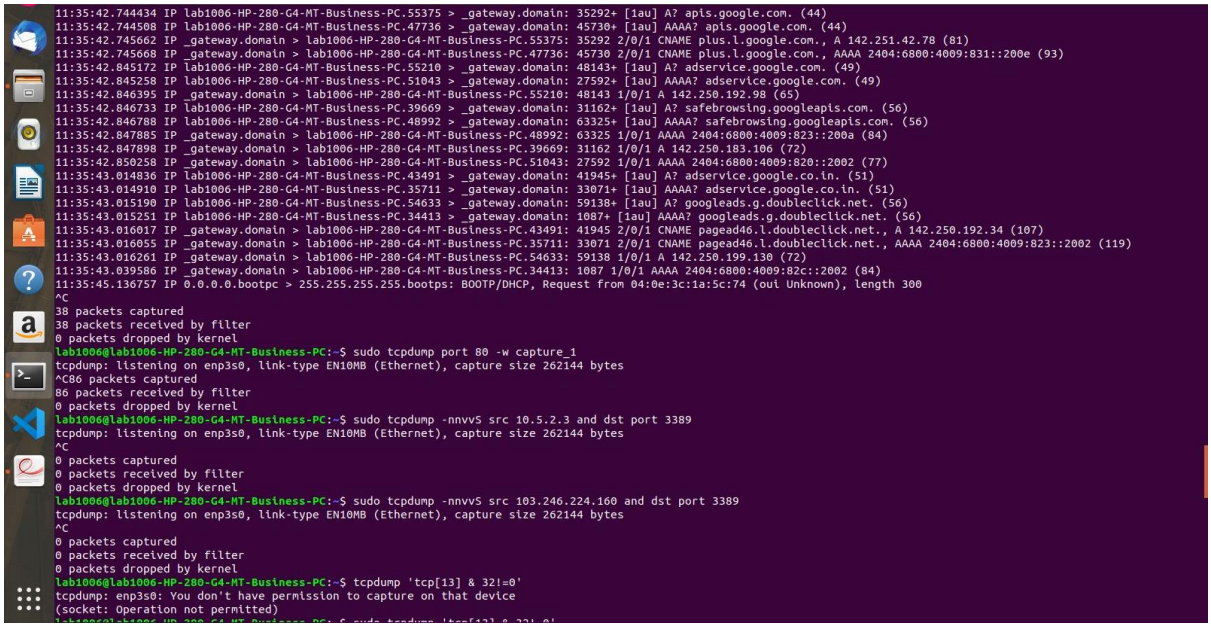
```
    192.168.0.115.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:06:48.559410 04:0e:3c:19:2d:d2 > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 74: (hlim 1, next-header UDP (17) payload length: 20) fe80::d08:56ec:5b45:e946.53
53 > ff02::fb.5353: [udp sum ok] 0*- [0q] 0/0/0 (12)
11:06:48.559669 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 1, id 17398, offset 0, flags [none], proto UDP (17), length 40)
    192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:06:48.859888 a4:ae:12:84:80:ea > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.134 tell 192.168.0.1
85, length 46
11:06:49.156505 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: (tos 0x0, ttl 1, id 620, offset 0, flags [none], proto UDP (17), length 204)
    192.44.44.202.60046 > 239.255.255.250.1900: UDP, length 176
11:06:49.943390 04:0e:3c:1a:60:2f > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.141 tell 192.168.0.1
90, length 46
11:06:50.170534 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: (tos 0x0, ttl 1, id 621, offset 0, flags [none], proto UDP (17), length 204)
    192.44.44.202.60046 > 239.255.255.250.1900: UDP, length 176
11:06:50.567535 04:0e:3c:1a:60:2f > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.141 tell 192.168.0.1
90, length 46
11:06:50.584752 04:0e:3c:1b:d1:42 > ac:15:a2:b9:9e:29, ethertype IPv4 (0x0800), length 105: (tos 0x0, ttl 64, id 15809, offset 0, flags [DF], proto TCP (6), length 91)
    192.168.0.213.51252 > 185.199.108.154.443: Flags [P.], cksum 0xe82c (incorrect -> 0xc33f), seq 1873020008:1873020047, ack 1011178678, win 4607, options [nop,nop,TS
val 3555857482 ecr 1680001507], length 39
11:06:50.601299 ac:15:a2:b9:9e:29 > 04:0e:3c:1b:d1:42, ethertype IPv4 (0x0800), length 66: (tos 0x34, ttl 55, id 60381, offset 0, flags [DF], proto TCP (6), length 52)
    185.199.108.154.443 > 192.168.0.213.51252: Flags [.], cksum 0xbab0 (correct), ack 39, win 284, options [nop,nop,TS val 1680060306 ecr 3555857482], length 0
11:06:50.601323 ac:15:a2:b9:9e:29 > 04:0e:3c:1b:d1:42, ethertype IPv4 (0x0800), length 105: (tos 0x34, ttl 55, id 60382, offset 0, flags [DF], proto TCP (6), length 91)
    185.199.108.154.443 > 192.168.0.213.51252: Flags [P.], cksum 0x5056 (correct), seq 1:40, ack 39, win 284, options [nop,nop,TS val 1680060306 ecr 3555857482], length
 39
11:06:50.601400 04:0e:3c:1b:d1:42 > ac:15:a2:b9:9e:29, ethertype IPv4 (0x0800), length 66: (tos 0x0, ttl 64, id 15810, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.213.51252 > 185.199.108.154.443: Flags [.], cksum 0xe805 (incorrect -> 0xa995), ack 40, win 4607, options [nop,nop,TS val 3555857499 ecr 1680060306], leng
th 0
11:06:50.675239 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 428: (tos 0x0, ttl 2, id 37639, offset 0, flags [DF], proto UDP (17), length 414)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 386
11:06:50.675463 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 437: (tos 0x0, ttl 2, id 37640, offset 0, flags [DF], proto UDP (17), length 423)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 395
11:06:50.675567 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 500: (tos 0x0, ttl 2, id 37641, offset 0, flags [DF], proto UDP (17), length 486)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 458
11:06:50.675836 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 496: (tos 0x0, ttl 2, id 37642, offset 0, flags [DF], proto UDP (17), length 482)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 454
11:06:50.675997 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 476: (tos 0x0, ttl 2, id 37643, offset 0, flags [DF], proto UDP (17), length 462)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 434
11:06:50.676187 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 508: (tos 0x0, ttl 2, id 37644, offset 0, flags [DF], proto UDP (17), length 494)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 466
11:06:50.676299 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 490: (tos 0x0, ttl 2, id 37645, offset 0, flags [DF], proto UDP (17), length 476)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 448
11:06:50.676493 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 492: (tos 0x0, ttl 2, id 37646, offset 0, flags [DF], proto UDP (17), length 478)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 450
11:06:50.676669 ac:15:a2:b9:9e:29 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 492: (tos 0x0, ttl 2, id 37647, offset 0, flags [DF], proto UDP (17), length 478)
    192.168.0.1.60033 > 239.255.255.250.1900: UDP, length 450
11:06:50.696063 04:0e:3c:1a:5c:1f > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 102: (flowlabel 0x061d2, hlim 255, next-header UDP (17) payload length: 48) fe80:
```

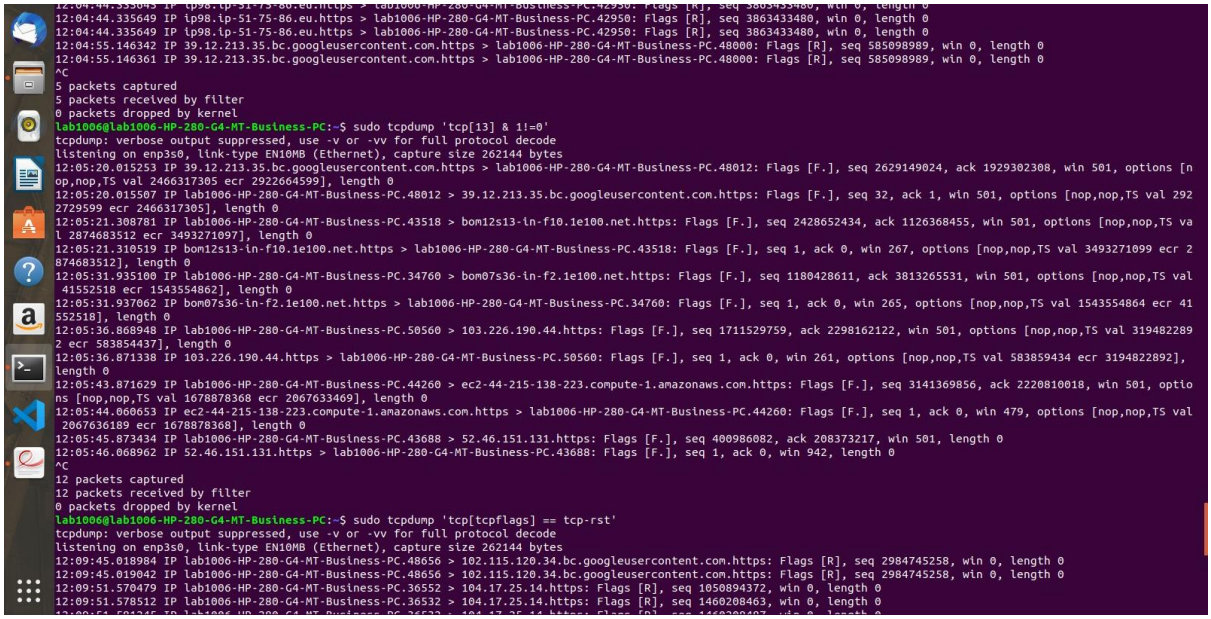


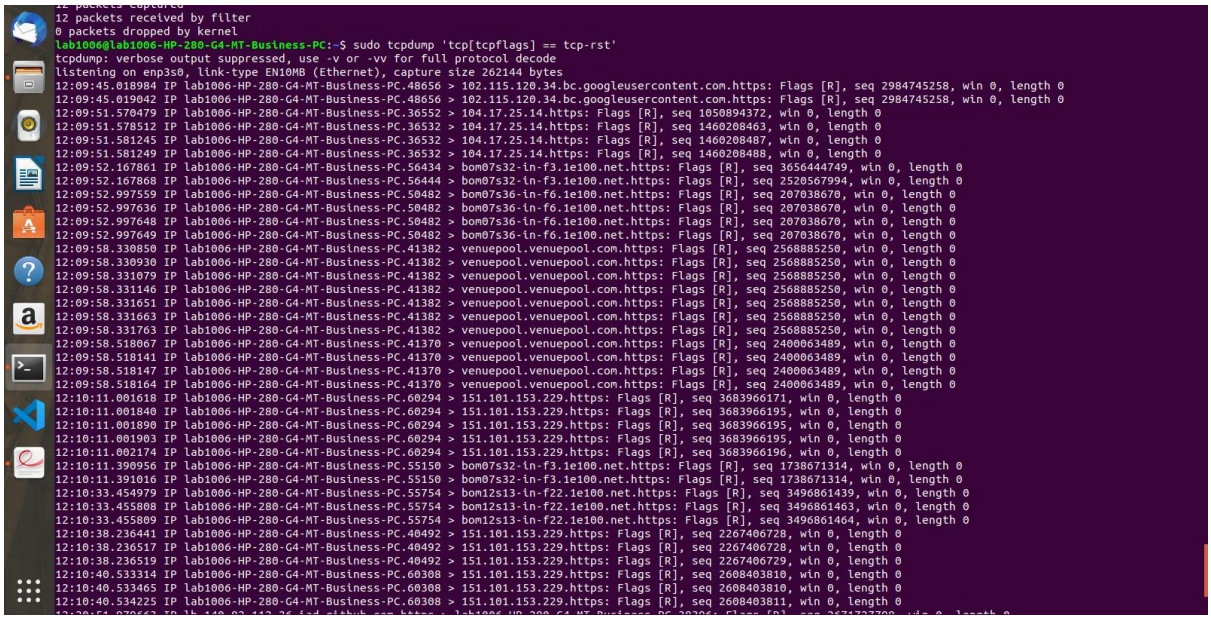
```
    192.168.0.213.38292 > 152.195.38.76.80: Flags [.], cksum 0x80b3 (incorrect -> 0xbd9d), ack 1018572014, win 501, options [nop,nop,TS val 1531651325 ecr 4184076819],
length 0
11:06:51.426290 ac:15:a2:b9:9e:29 > 04:0e:3c:1b:d1:42, ethertype IPv4 (0x0800), length 66: (tos 0x0, ttl 58, id 61638, offset 0, flags [none], proto TCP (6), length 52)
    152.195.38.76.80 > 192.168.0.213.38292: Flags [.], cksum 0xeb9a (correct), ack 1, win 135, options [nop,nop,TS val 4184087059 ecr 1531629677], length 0
11:06:51.567263 04:0e:3c:1a:60:2f > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.141 tell 192.168.0.1
90, length 46
11:06:51.691627 a4:ae:12:84:80:ea > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.134 tell 192.168.0.1
85, length 46
11:06:51.831366 ac:15:a2:b9:9e:29 > 04:0e:3c:1b:d1:42, ethertype IPv4 (0x0800), length 91: (tos 0x34, ttl 46, id 7178, offset 0, flags [DF], proto TCP (6), length 77)
    140.82.112.25.443 > 192.168.0.213.37992: Flags [P.], cksum 0xdcb4 (correct), seq 3601914876:3601914901, ack 4276782204, win 77, options [nop,nop,TS val 3554971416 e
cr 3093326561], length 25
11:06:51.831411 04:0e:3c:1b:d1:42 > ac:15:a2:b9:9e:29, ethertype IPv4 (0x0800), length 66: (tos 0x0, ttl 64, id 46494, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.213.37992 > 140.82.112.25.443: Flags [.], cksum 0xbe0f (incorrect -> 0xc80d), ack 25, win 501, options [nop,nop,TS val 3093366561 ecr 3554971416], length
0
11:06:51.831638 04:0e:3c:1b:d1:42 > ac:15:a2:b9:9e:29, ethertype IPv4 (0x0800), length 95: (tos 0x0, ttl 64, id 46495, offset 0, flags [DF], proto TCP (6), length 81)
    192.168.0.213.37992 > 140.82.112.25.443: Flags [P.], cksum 0xbe2c (incorrect -> 0x50e9), seq 1:30, ack 25, win 501, options [nop,nop,TS val 3093366561 ecr 355497141
6], length 29
^C
43 packets captured
43 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:08:51.831101 IP 140.82.112.25.443 > 192.168.0.213.37992: Flags [P.], seq 3601914926:3601914951, ack 4276782262, win 77, options [nop,nop,TS val 3555091411 ecr 309342
6561], length 25
11:08:51.831308 IP 192.168.0.213.37992 > 140.82.112.25.443: Flags [P.], seq 1:30, ack 25, win 501, options [nop,nop,TS val 3093486561 ecr 3555091411], length 29
11:08:52.119414 IP 140.82.112.25.443 > 192.168.0.213.37992: Flags [.], ack 30, win 77, options [nop,nop,TS val 3555091700 ecr 3093486561], length 0
11:08:57.438567 IP 192.168.0.213.39510 > 140.82.114.22.443: Flags [.], seq 2871837009:2871837045, ack 1471998315, win 501, options [nop,nop,TS val 3554701205 ecr 242720
383], length 36
11:08:57.438586 IP 192.168.0.213.39510 > 140.82.114.22.443: Flags [P.], seq 36:39, ack 1, win 501, options [nop,nop,TS val 3554701205 ecr 242720383], length 3
11:08:57.438660 IP 192.168.0.213.51198 > 185.199.108.154.443: Flags [.], seq 4064335366:4064335402, ack 768197512, win 11904, options [nop,nop,TS val 3555984336 ecr 130
5689667], length 36
11:08:57.438664 IP 192.168.0.213.51198 > 185.199.108.154.443: Flags [P.], seq 36:39, ack 1, win 11904, options [nop,nop,TS val 3555984336 ecr 1305689667], length 3
11:08:57.439030 IP 192.168.0.213.51198 > 185.199.108.154.443: Flags [P.], seq 39:63, ack 1, win 11904, options [nop,nop,TS val 3555984337 ecr 1305689667], length 24
11:08:57.439057 IP 192.168.0.213.51198 > 185.199.108.154.443: Flags [F.], seq 63, ack 1, win 11904, options [nop,nop,TS val 3555984337 ecr 1305689667], length 0
11:08:57.439775 IP 192.168.0.213.39510 > 140.82.114.22.443: Flags [P.], seq 39:63, ack 1, win 501, options [nop,nop,TS val 3554701206 ecr 242720383], length 24
11:08:57.439800 IP 192.168.0.213.39510 > 140.82.114.22.443: Flags [F.], seq 63, ack 1, win 501, options [nop,nop,TS val 3554701206 ecr 242720383], length 0
11:08:57.455029 IP 185.199.108.154.443 > 192.168.0.213.51198: Flags [.], ack 0, win 377, options [nop,nop,TS val 1305742675 ecr 3555931344,nop,nop,sack 1 {36:39}], leng
th 0
11:08:57.455044 IP 185.199.108.154.443 > 192.168.0.213.51198: Flags [.], ack 39, win 377, options [nop,nop,TS val 1305742675 ecr 3555984336], length 0
11:08:57.457877 IP 185.199.108.154.443 > 192.168.0.213.51198: Flags [.], ack 63, win 377, options [nop,nop,TS val 1305742675 ecr 3555984337], length 0
11:08:57.457892 IP 185.199.108.154.443 > 192.168.0.213.51198: Flags [P.], seq 1:25, ack 63, win 377, options [nop,nop,TS val 1305742675 ecr 3555984337], length 24
11:08:57.457894 IP 185.199.108.154.443 > 192.168.0.213.51198: Flags [F.], seq 25, ack 63, win 377, options [nop,nop,TS val 1305742675 ecr 3555984337], length 0
11:08:57.457958 IP 192.168.0.213.51198 > 185.199.108.154.443: Flags [R], seq 4064335429, win 0, length 0
```


25 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n tcp src 192.168.0.181
tcpdump: 'tcp' modifier applied to host
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n src 192.168.0.181 icmp
tcpdump: syntax error in filter expression: syntax error
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp src 192.168.0.181 icmp
tcpdump: 'icmp' modifier applied to host
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:23:14.623598 IP 192.168.0.213 > 103.246.224.160: ICMP echo request, id 13898, seq 10, length 64
11:23:14.624221 IP 103.246.224.160 > 192.168.0.213: ICMP echo reply, id 13898, seq 10, length 64
11:23:15.647605 IP 192.168.0.213 > 103.246.224.160: ICMP echo request, id 13898, seq 11, length 64
11:23:15.648227 IP 103.246.224.160 > 192.168.0.213: ICMP echo reply, id 13898, seq 11, length 64
11:23:16.671565 IP 192.168.0.213 > 103.246.224.160: ICMP echo request, id 13898, seq 12, length 64
11:23:16.672192 IP 103.246.224.160 > 192.168.0.213: ICMP echo reply, id 13898, seq 12, length 64
11:23:17.695594 IP 192.168.0.213 > 103.246.224.160: ICMP echo request, id 13898, seq 13, length 64
11:23:17.696161 IP 103.246.224.160 > 192.168.0.213: ICMP echo reply, id 13898, seq 13, length 64
11:23:18.719632 IP 192.168.0.213 > 103.246.224.160: ICMP echo request, id 13898, seq 14, length 64
11:23:18.720145 IP 103.246.224.160 > 192.168.0.213: ICMP echo reply, id 13898, seq 14, length 64
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcp port 80
sudo: tcp: command not found
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:28:38.285039 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3903811227, win 64240, options [mss 1460,sackOK,
TS val 3444133253 ecr 0,nop,wscale 7], length 0
11:28:39.295561 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3903811227, win 64240, options [mss 1460,sackOK,
TS val 3444134263 ecr 0,nop,wscale 7], length 0
11:28:39.538360 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [S.], seq 1089476767, ack 3903811228, win 64768, options
[mss 1420,sackOK,TS val 1089564564 ecr 3444134263,nop,wscale 7], length 0
11:28:39.538421 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [.], ack 1, win 502, options [nop,nop,TS val 3444134506 e
cr 1089564564], length 0


11:28:39.941579 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [F.], seq 149, ack 88, win 506, options [nop,nop,TS val 1
089565016 ecr 3444134506], length 0
11:28:39.941608 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [.], ack 150, win 501, options [nop,nop,TS val 3444134909
ecr 1089565016], length 0
11:28:40.183386 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [.], ack 89, win 506, options [nop,nop,TS val 1089565258
ecr 3444134908], length 0
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump udp and src port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:33:37.241511 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.57215: 10986 9/3/1 A 34.122.121.32, A 35.224.170.84, A 185.125.190.18, A 35.232.111.17, A 91.189.9
1.48, A 185.125.190.49, A 185.125.190.17, A 91.189.91.49, A 185.125.190.40 (266)
11:33:37.241594 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.32907: 3486 6/3/1 AAAA 2001:67c:1562::23, AAAA 2620:2d:4000:1::23, AAAA 2620:2d:4000:1::2b, AAAA 2
620:2d:4000:1::22, AAAA 2001:67c:1562::24, AAAA 2620:2d:4000:1::2a (290)
11:34:04.686194 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.53528: 54238 4/4/1 A 108.158.61.90, A 108.158.61.4, A 108.158.61.10, A 108.158.61.13 (258)
11:34:04.709453 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.59252: 37086 8/4/1 AAAA 2600:9000:237b:c200:1a:5235:f980:93a1, AAAA 2600:9000:237b:c000:1a:5235:f9
80:93a1, AAAA 2600:9000:237b:400:1a:5235:f980:93a1, AAAA 2600:9000:237b:d400:1a:5235:f980:93a1, AAAA 2600:9000:237b:7800:1a:5235:f980:93a1, AAAA 2600:9000:237b:7e00:1a:
5235:f980:93a1, AAAA 2600:9000:237b:dc00:1a:5235:f980:93a1, AAAA 2600:9000:237b:2800:1a:5235:f980:93a1 (418)
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump portrange 1-80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:35:13.653873 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74 (oui Unknown), length 300
11:35:17.801654 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74 (oui Unknown), length 300
11:35:22.173999 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74 (oui Unknown), length 300
11:35:30.078393 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74 (oui Unknown), length 300
11:35:38.922635 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 1c:6f:65:ae:98:2a (oui Unknown), length 300
11:35:41.918550 IP lab1006-HP-280-G4-MT-Business-PC.36586 > _gateway.domain: 53847+ [1au] A? encrypted-tbn0.gstatic.com. (55)
11:35:41.918818 IP lab1006-HP-280-G4-MT-Business-PC.35381 > _gateway.domain: 12276+ [1au] AAAA? encrypted-tbn0.gstatic.com. (55)
11:35:41.919849 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.36586: 53847 1/0/1 A 142.250.183.78 (71)
11:35:41.938280 IP lab1006-HP-280-G4-MT-Business-PC.56668 > _gateway.domain: 933+ [1au] A? www.google.com. (43)
11:35:41.938421 IP lab1006-HP-280-G4-MT-Business-PC.59077 > _gateway.domain: 26727+ [1au] AAAA? www.google.com. (43)
11:35:41.939510 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.56668: 933 1/0/1 A 172.217.27.196 (59)
11:35:41.939601 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.59077: 26727 1/0/1 AAAA 2404:6800:4009:800::2004 (71)
11:35:41.980589 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.35381: 12276 1/0/1 AAAA 2404:6800:4009:822::200e (83)
11:35:42.677951 IP lab1006-HP-280-G4-MT-Business-PC.37545 > _gateway.domain: 56141+ [1au] A? www.gstatic.com. (44)
11:35:42.678020 IP lab1006-HP-280-G4-MT-Business-PC.41726 > _gateway.domain: 30891+ [1au] AAAA? www.gstatic.com. (44)
11:35:42.679208 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.41726: 30891 1/0/1 AAAA 2404:6800:4009:82b::2003 (72)
11:35:42.679329 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.37545: 56141 1/0/1 A 142.250.192.131 (60)


11:35:42.744434 IP lab1006-HP-280-G4-MT-Business-PC.55375 > _gateway.domain: 35292+ [1au] A? apis.google.com. (44)
11:35:42.744508 IP lab1006-HP-280-G4-MT-Business-PC.47736 > _gateway.domain: 45730+ [1au] AAAA? apis.google.com. (44)
11:35:42.745662 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.55375: 35292 2/0/1 CNAME plus.l.google.com., A 142.251.42.78 (81)
11:35:42.745668 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.47736: 45730 2/0/1 CNAME plus.l.google.com., AAAA 2404:6800:4009:831::200e (93)
11:35:42.845172 IP lab1006-HP-280-G4-MT-Business-PC.55210 > _gateway.domain: 48143+ [1au] A? adservice.google.com. (49)
11:35:42.845258 IP lab1006-HP-280-G4-MT-Business-PC.51043 > _gateway.domain: 27592+ [1au] AAAA? adservice.google.com. (49)
11:35:42.846395 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.55210: 48143 1/0/1 A 142.250.192.98 (65)
11:35:42.846733 IP lab1006-HP-280-G4-MT-Business-PC.39669 > _gateway.domain: 31162+ [1au] A? safebrowsing.googleapis.com. (56)
11:35:42.846788 IP lab1006-HP-280-G4-MT-Business-PC.48992 > _gateway.domain: 63325+ [1au] AAAA? safebrowsing.googleapis.com. (56)
11:35:42.847885 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.48992: 63325 1/0/1 AAAA 2404:6800:4009:823::200a (84)
11:35:42.847898 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.39669: 31162 1/0/1 A 142.250.183.106 (72)
11:35:42.850258 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.51043: 27592 1/0/1 AAAA 2404:6800:4009:820::2002 (77)
11:35:43.014836 IP lab1006-HP-280-G4-MT-Business-PC.43491 > _gateway.domain: 41945+ [1au] A? adservice.google.co.in. (51)
11:35:43.014910 IP lab1006-HP-280-G4-MT-Business-PC.35711 > _gateway.domain: 33071+ [1au] AAAA? adservice.google.co.in. (51)
11:35:43.015190 IP lab1006-HP-280-G4-MT-Business-PC.54633 > _gateway.domain: 59138+ [1au] A? googleads.g.doubleclick.net. (56)
11:35:43.015251 IP lab1006-HP-280-G4-MT-Business-PC.34413 > _gateway.domain: 1087+ [1au] AAAA? googleads.g.doubleclick.net. (56)
11:35:43.016017 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.43491: 41945 2/0/1 CNAME pagead46.l.doubleclick.net., A 142.250.192.34 (107)
11:35:43.016055 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.35711: 33071 2/0/1 CNAME pagead46.l.doubleclick.net., AAAA 2404:6800:4009:823::2002 (119)
11:35:43.016261 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.54633: 59138 1/0/1 A 142.250.199.130 (72)
11:35:43.039586 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.34413: 1087 1/0/1 AAAA 2404:6800:4009:82c::2002 (84)
11:35:45.136757 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74 (oui Unknown), length 300
^C
38 packets captured
38 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump port 80 -w capture_1
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C86 packets captured
86 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -nnvvS src 10.5.2.3 and dst port 3389
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -nnvvS src 103.246.224.160 and dst port 3389
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tcpdump 'tcp[13] & 32!=0'
tcpdump: enp3s0: You don't have permission to capture on that device
(socket: Operation not permitted)





CONCLUSION:

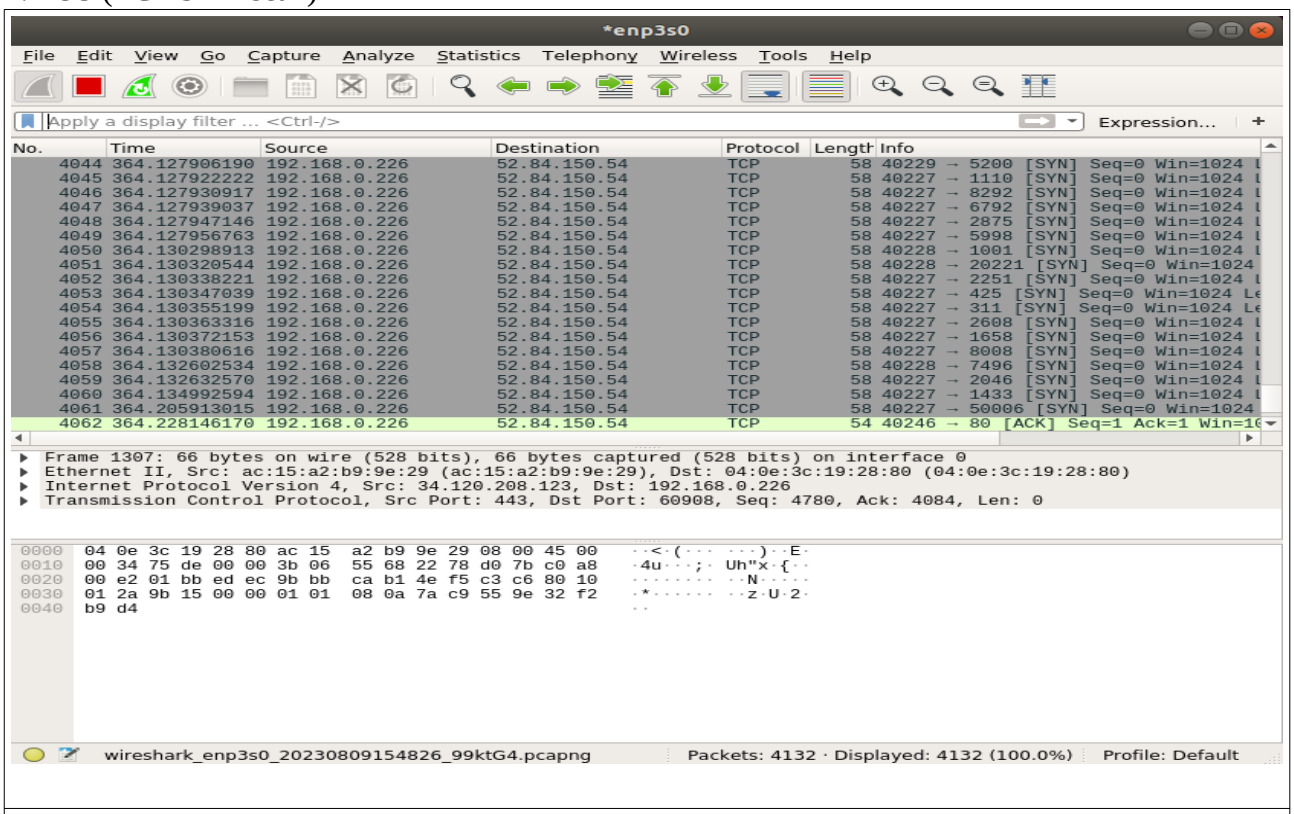
We gained a practical understanding of how TCPDump can be employed to capture, dissect, and interpret network packets in real-time, offering valuable insights into network behavior, troubleshooting, and security assessment. By applying various filters and commands, we were able to capture specific types of traffic based on source and destination addresses, protocols, ports, and packet sizes.

Aim: Installation of NMAP and using it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP port scan, etc

LO mapped: LO4

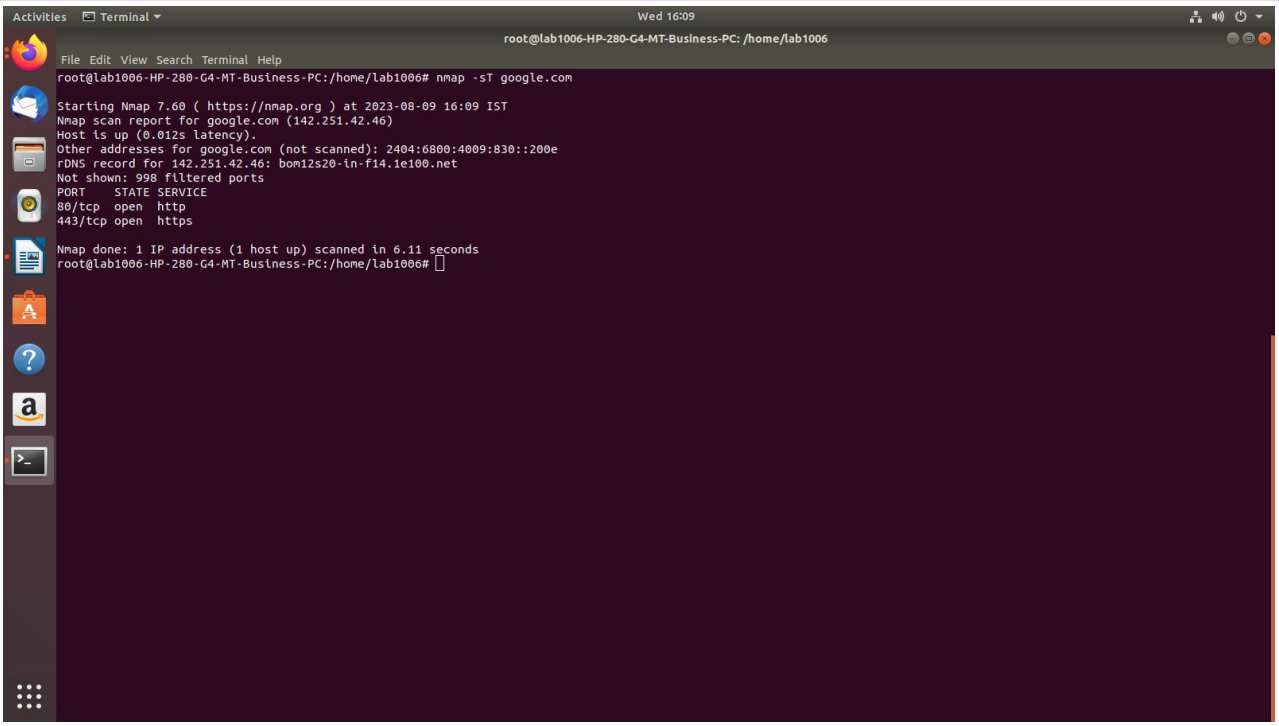
Ping Sweep
Nmap -sP <IP address(192.168.0.*)>

1. `-sS` (TCP SYN scan)



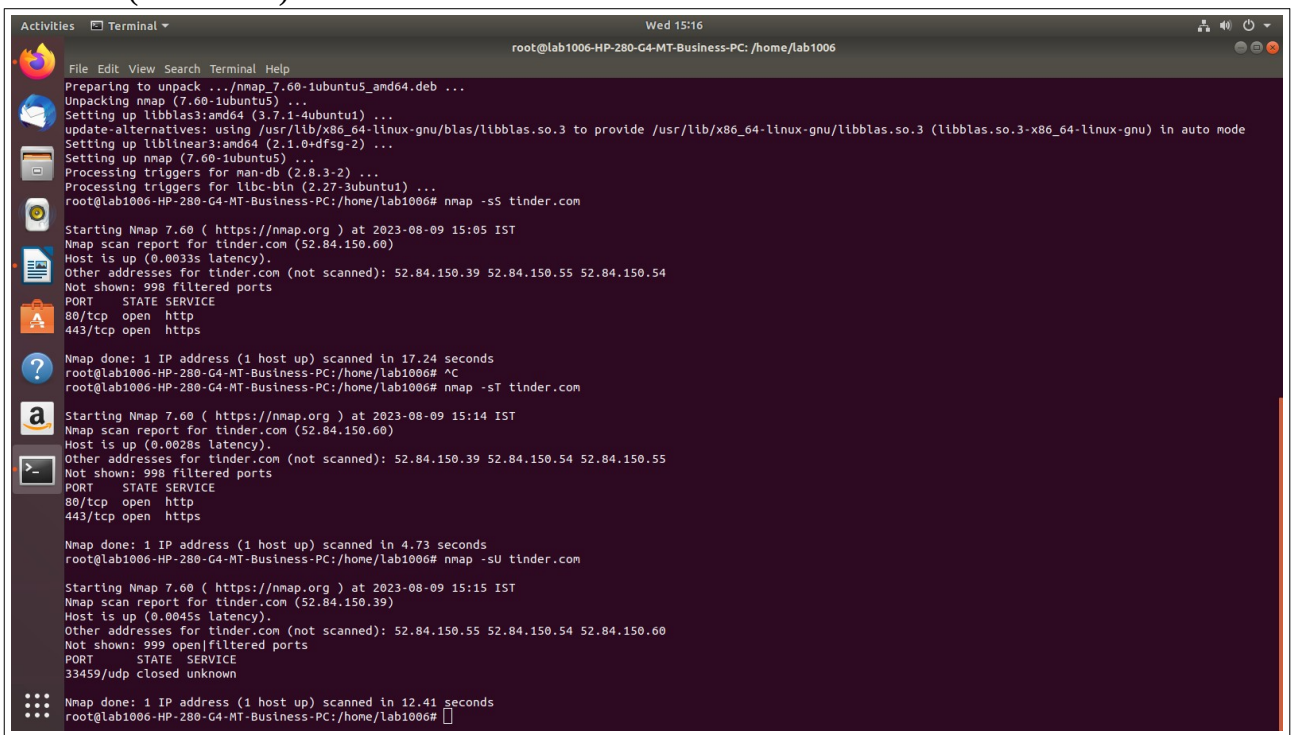
SYN scan is the default and most popular scan option for good reason. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by intrusive firewalls. SYN scan is relatively unobtrusive and stealthy, since it never completes TCP connections. It also works against any compliant TCP stack rather than depending on idiosyncrasies of specific platforms as Nmap's FIN/NULL/Xmas, Maimon and idle scans do. It also allows clear, reliable differentiation between `open`, `closed`, and `filtered` states.

2. `-sT` (TCP connect scan)



TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges or is scanning IPv6 networks. Instead of writing raw packets as most other scan types do, Nmap asks the underlying operating system to establish a connection with the target machine and port by issuing the `connect` system call. This is the same high-level system call that web browsers, P2P clients, and most other network-enabled applications use to establish a connection. It is part of a programming interface known as the Berkeley Sockets API. Rather than read raw packet responses off the wire, Nmap uses this API to obtain status information on each connection attempt. This and the FTP bounce scan (the section called “TCP FTP Bounce Scan (`-b`)”) are the only scan types available to unprivileged users.

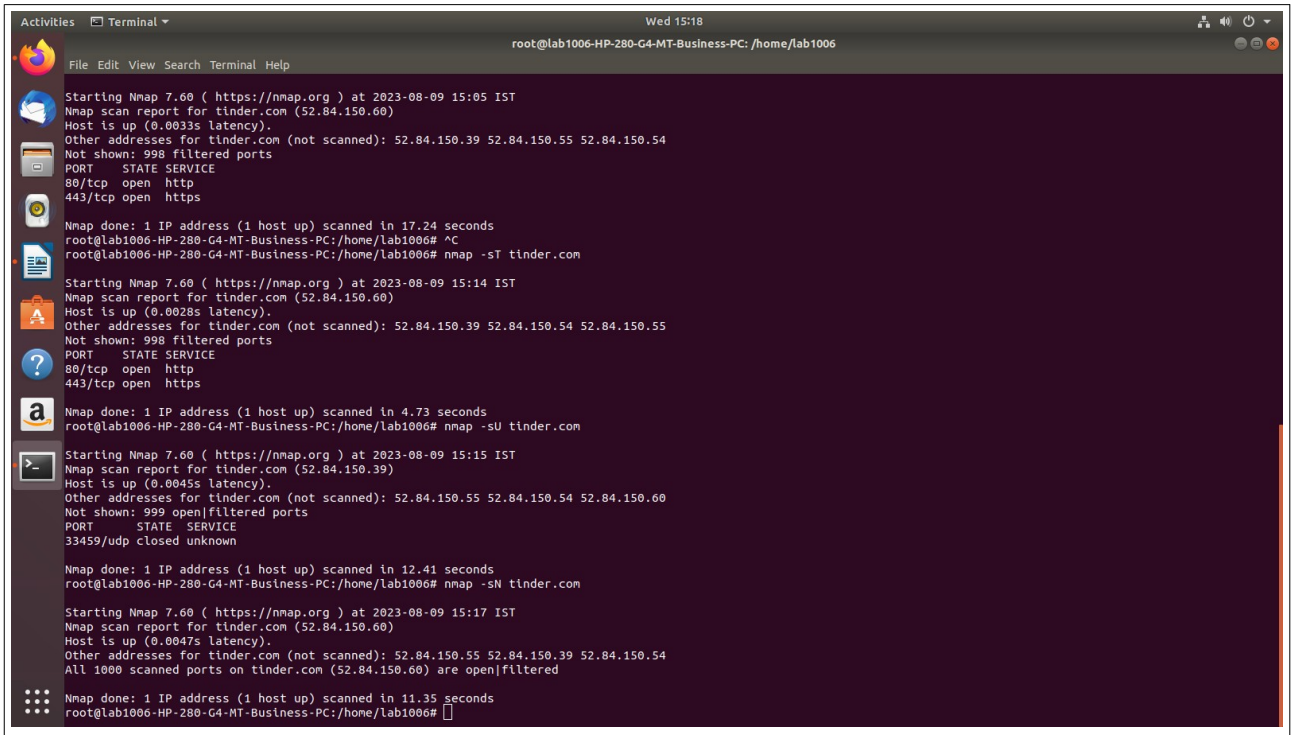
3. `-sU` (UDP scans)

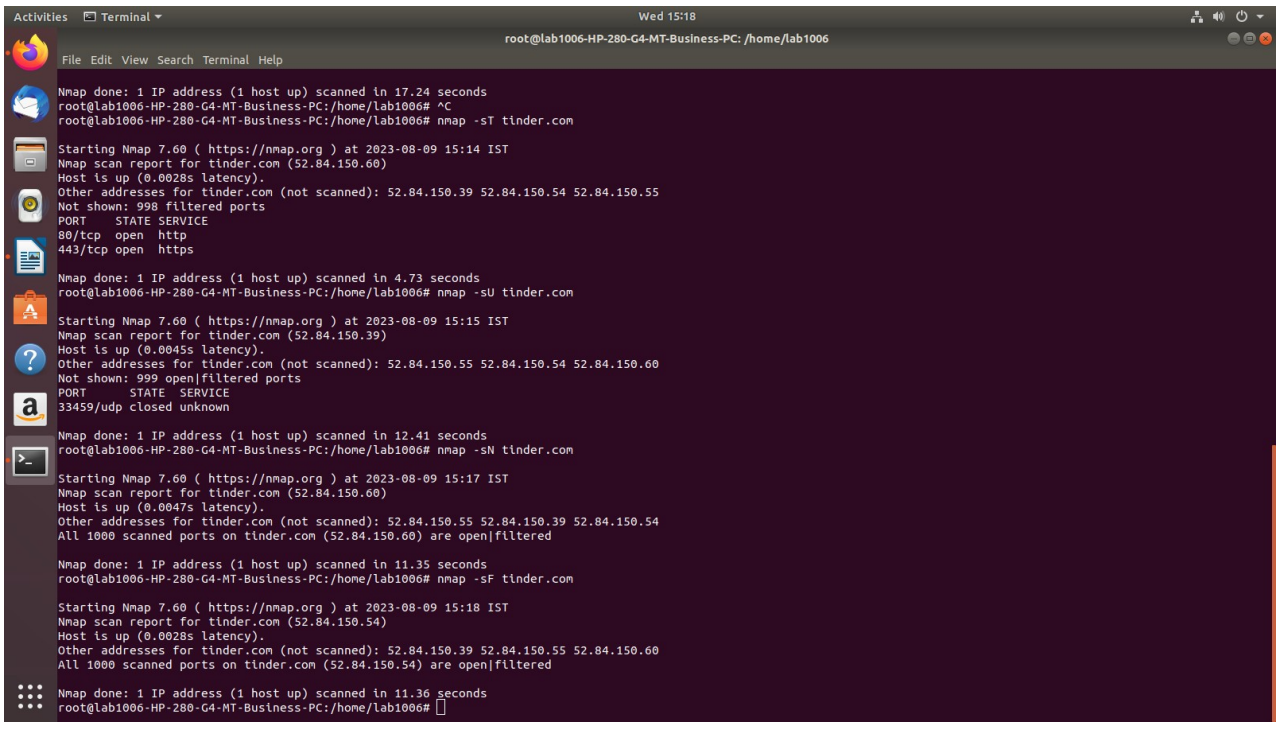


While most popular services on the Internet run over the TCP protocol, UDP services are widely deployed. DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common. Because UDP scanning is generally slower and more difficult than TCP, some security auditors ignore these ports. This is a mistake, as exploitable UDP services are quite common and attackers certainly don't ignore the whole protocol. Fortunately, Nmap can help inventory UDP ports.

UDP scan is activated with the `-sU` option. It can be combined with a TCP scan type such as SYN scan (`-sS`) to check both protocols during the same run.

`4.-sN`; `-sF`; `-sX` (TCP NULL, FIN, and Xmas scans)


Activities
Terminal
Wed 15:18
root@lab1006-HP-280-G4-MT-Business-PC: /home/lab1006
File Edit View Search Terminal Help
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:05 IST
Nmap scan report for tinder.com (52.84.150.60)
Host is up (0.0033s latency).
Other addresses for tinder.com (not scanned): 52.84.150.39 52.84.150.55 52.84.150.54
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 17.24 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# ^C
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sT tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:14 IST
Nmap scan report for tinder.com (52.84.150.60)
Host is up (0.0028s latency).
Other addresses for tinder.com (not scanned): 52.84.150.39 52.84.150.54 52.84.150.55
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 4.73 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sU tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:15 IST
Nmap scan report for tinder.com (52.84.150.39)
Host is up (0.0045s latency).
Other addresses for tinder.com (not scanned): 52.84.150.55 52.84.150.54 52.84.150.60
Not shown: 999 open|filtered ports
PORT STATE SERVICE
33459/udp closed unknown
Nmap done: 1 IP address (1 host up) scanned in 12.41 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sN tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:17 IST
Nmap scan report for tinder.com (52.84.150.60)
Host is up (0.0047s latency).
Other addresses for tinder.com (not scanned): 52.84.150.55 52.84.150.39 52.84.150.54
All 1000 scanned ports on tinder.com (52.84.150.60) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 11.35 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#


Activities
Terminal
Wed 15:18
root@lab1006-HP-280-G4-MT-Business-PC: /home/lab1006
File Edit View Search Terminal Help
Nmap done: 1 IP address (1 host up) scanned in 17.24 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# ^C
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sT tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:14 IST
Nmap scan report for tinder.com (52.84.150.60)
Host is up (0.0028s latency).
Other addresses for tinder.com (not scanned): 52.84.150.39 52.84.150.54 52.84.150.55
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 4.73 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sU tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:15 IST
Nmap scan report for tinder.com (52.84.150.39)
Host is up (0.0045s latency).
Other addresses for tinder.com (not scanned): 52.84.150.55 52.84.150.54 52.84.150.60
Not shown: 999 open|filtered ports
PORT STATE SERVICE
33459/udp closed unknown
Nmap done: 1 IP address (1 host up) scanned in 12.41 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sN tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:17 IST
Nmap scan report for tinder.com (52.84.150.60)
Host is up (0.0047s latency).
Other addresses for tinder.com (not scanned): 52.84.150.55 52.84.150.39 52.84.150.54
All 1000 scanned ports on tinder.com (52.84.150.60) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 11.35 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sF tinder.com
Starting Nmap 7.60 (https://nmap.org) at 2023-08-09 15:18 IST
Nmap scan report for tinder.com (52.84.150.54)
Host is up (0.0028s latency).
Other addresses for tinder.com (not scanned): 52.84.150.39 52.84.150.55 52.84.150.60
All 1000 scanned ports on tinder.com (52.84.150.54) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#

These three scan types (even more are possible with the `--scanflags` option described in the next section) exploit a subtle loophole in the TCP RFC to differentiate between `open` and `closed` ports. Page 65 of RFC 793 says that “if the [destination] port state is CLOSED .... an incoming segment not containing a RST causes a RST to be sent in response.” Then the next page discusses packets sent to open ports without the SYN, RST, or ACK bits set, stating that: “you are unlikely to get here, but if you do, drop the segment, and return.”

When scanning systems compliant with this RFC text, any packet not containing SYN, RST, or ACK bits will result in a returned RST if the port is closed and no response at all if the port is open. As long as none of those three bits are included, any combination of the other three (FIN, PSH, and URG) are OK. Nmap exploits this with three scan types:

Null scan (`-sN`)

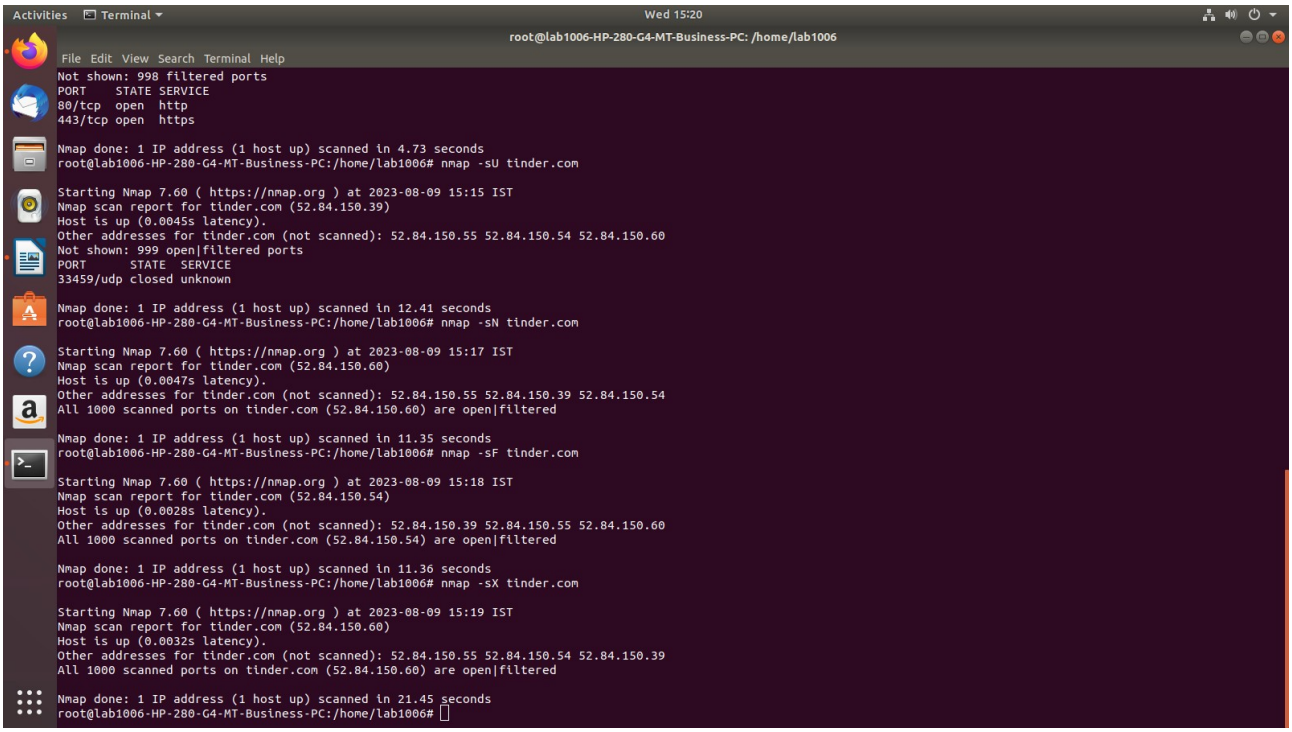
Does not set any bits (TCP flag header is 0)

FIN scan (`-sF`)

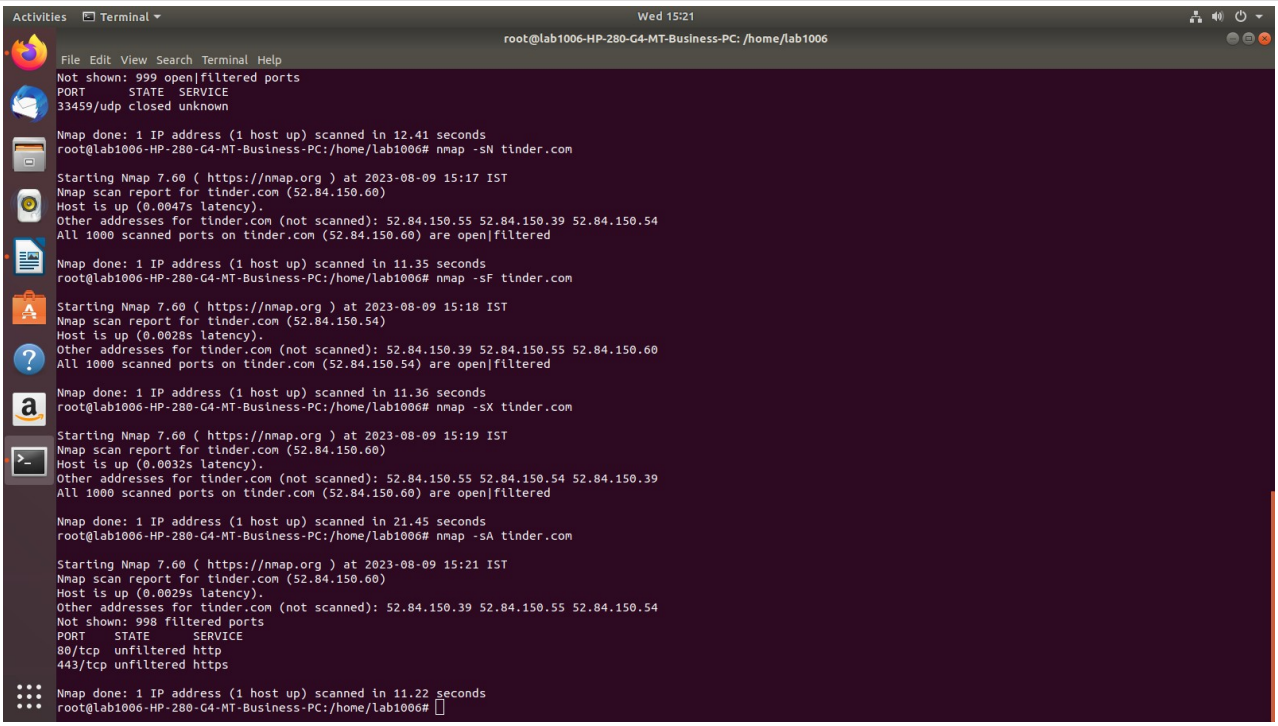
Sets just the TCP FIN bit.

Xmas scan (`-sX`)

Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.



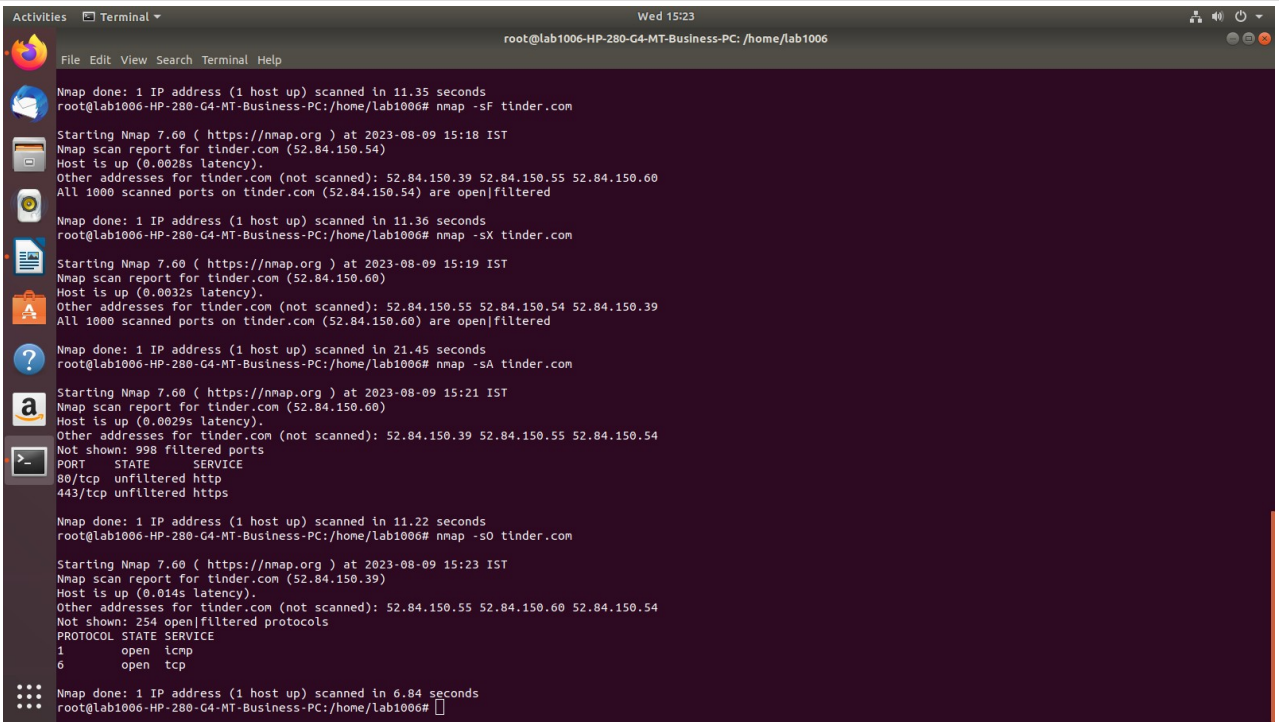
5. `-sA` (TCP ACK scan)



This scan is different than the others discussed so far in that it never determines `open` (or even `open|filtered`) ports. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.

ACK scan is enabled by specifying the `-sA` option. Its probe packet has only the ACK flag set (unless you use `--scanflags`). When scanning unfiltered systems, `open` and `closed` ports will both return a RST packet. Nmap then labels them as `unfiltered`, meaning that they are reachable by the ACK packet, but whether they are `open` or `closed` is undetermined. Ports that don't respond, or send certain ICMP error messages back, are labeled `filtered`. Table 5.5 provides the full details.

6. `-sO (IP protocol scan)`



IP protocol scan allows you to determine which IP protocols (TCP, ICMP, IGMP, etc.) are supported by target machines. This isn't technically a port scan, since it cycles through IP protocol numbers rather than TCP or UDP port numbers. Yet it still uses the `-p` option to select scanned protocol numbers, reports its results within the normal port table format, and even uses the same underlying scan engine as the true port scanning methods. So it is close enough to a port scan that it belongs here.

Besides being useful in its own right, protocol scan demonstrates the power of open-source software. While the fundamental idea is pretty simple, I had not thought to add it nor received any requests for such functionality. Then in the summer of 2000, Gerhard Rieger conceived the idea, wrote an excellent patch implementing it, and sent it to the *nmap-hackers* mailing list. I incorporated that patch into the Nmap tree and released a new version the next day. Few pieces of commercial software have users enthusiastic enough to design and contribute their own improvements!

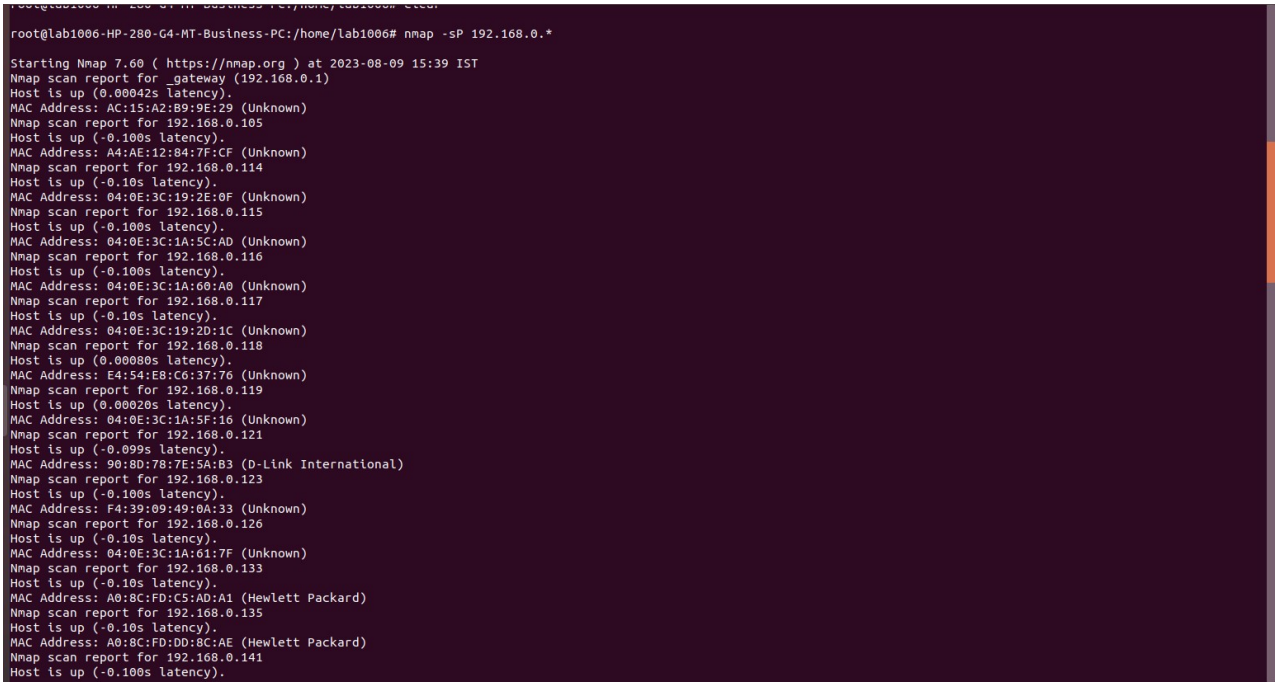
7. `-O` (Enable OS detection)

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -O 192.168.0.119

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:36 IST
Nmap scan report for 192.168.0.119
Host is up (0.00029s latency).
All 1000 scanned ports on 192.168.0.119 are closed
MAC Address: 04:0E:3C:1A:5F:16 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.66 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting.

Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its `nmap-os-db` database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match. Each fingerprint includes a freeform textual description of the OS, and a classification which provides the vendor name (e.g. Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (general purpose, router, switch, game console, etc). Most fingerprints also have a Common Platform Enumeration (CPE) representation, like `cpe:/o:linux:linux_kernel:2.6`.

8. nmap -sP 192.168.0.*



A ping sweep (also known as an ICMP sweep) is a basic network scanning technique used to determine which of a range of IP addresses map to live hosts (computers).

Whereas a single ping will tell whether one specified host computer exists on the network, a ping sweep consists of ICMP (Internet Control Message Protocol) *echo requests* sent to multiple hosts. To do this, the ping requires an address to send the echo request to, which can be an IP address or a web server domain name.

If a given address is live, it will return an ICMP *echo reply*. To disable ping sweeps on a network, administrators can block ICMP *echo requests* from outside sources. However, ICMP *timestamp* and *Address Mask requests* can be used in a similar manner.

Conclusion: By this assignment we implemented various different nmap network scanning commands and used wireshark.

Roll Number:- 122
Name:-Moheet Shendarkar
Date:- 06/09/2023

Lab Assignment No:-9

Aim:- Simulate DOS attack using HPING3.

Lab Outcome Attained :- LO5

Theory:-

What is Denial of Service Atack?

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a computer system, network, or online service by overwhelming it with a flood of illegitimate requests or traffic. The primary goal of a DoS attack is to make a resource, such as a website or server, unavailable to its intended users. It does so by consuming the target's resources, such as bandwidth, processing power, or memory, to the point where it cannot handle legitimate requests.

Explain SYN flood, ICMP flood and SMURF attack.

Three common types of DoS attacks:

SYN Flood Attack:

A SYN flood attack is a type of network-based DoS attack that targets the three-way handshake process in the Transmission Control Protocol (TCP), which is used for establishing connections between devices on the internet.

In a TCP connection, the client sends a SYN (synchronize) packet to initiate a connection with a server. The server is expected to respond with a SYN-ACK (synchronizeacknowledgment) packet, and then the client responds with an ACK (acknowledgment) packet to complete the handshake and establish the connection.

In a SYN flood attack, the attacker sends a high volume of SYN packets to the target server, but they do not complete the handshake by sending the expected ACK packets. This leaves the server waiting for the final ACKs, tying up its resources and preventing it from accepting legitimate connections.

SYN flood attacks can quickly overwhelm a server's ability to handle incoming connections, leading to service disruption.

ICMP Flood Attack:
An ICMP (Internet Control Message Protocol) flood attack, also known as a "ping flood" attack, targets the ICMP protocol, which is used for network diagnostics, particularly the "ping" command.

In this type of attack, the attacker sends a high volume of ICMP echo requests (ping requests) to the target system. Each request typically generates a response from the target, creating a flood of traffic.

ICMP flood attacks can consume the target's network bandwidth and processing resources, making it difficult for legitimate network traffic to pass through. This results in network congestion and service degradation or unavailability.

SMURF Attack:
A SMURF attack is a network-based DoS attack that takes advantage of ICMP and IP addressing.
In a SMURF attack, the attacker sends a large number of ICMP echo request (ping) packets to an IP broadcast address, typically spoofing the source IP address to make it appear as if the requests are coming from the victim's IP address.
When these requests are sent to the broadcast address, all devices on the target network respond with ICMP echo replies. With a high enough volume of requests, this can flood the victim's network, overwhelming its resources and causing a DoS.
To mitigate DoS attacks, organizations use various security measures, including firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and content delivery networks (CDNs). These tools help identify and filter out malicious traffic, allowing legitimate traffic to reach its destination. Additionally, proper network design and configuration can help minimize the impact of DoS attacks.

Write the Hping3 commands used for performing SYN flood and ICMP flood.

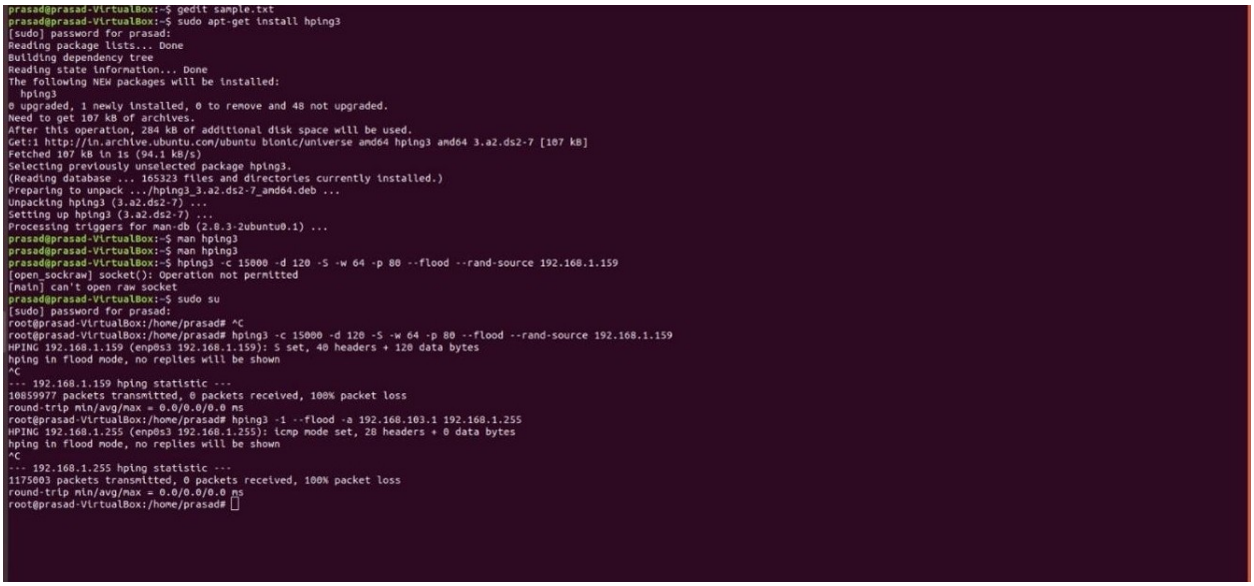
Syn flood :

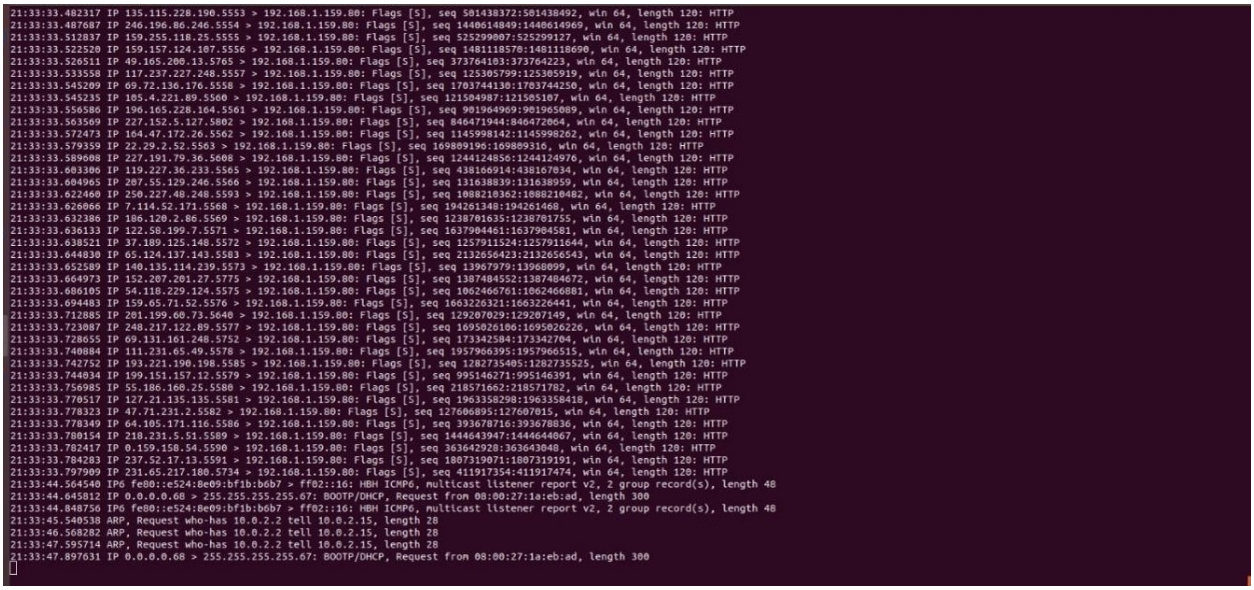
hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159

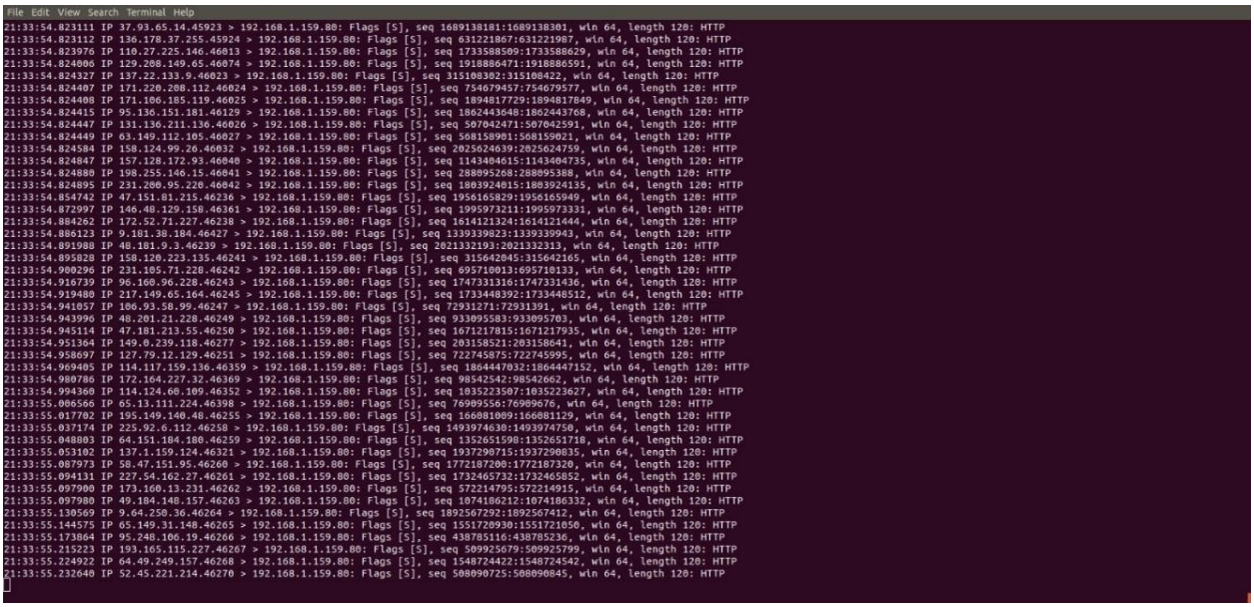
ICMP flood:

hping3 -1 --flood -a 192.168.103 192.168.1.255

Output Screenshots:-







Conclusion:-Learnt more about the network analysis and security assessment tools. Explored various network probing and testing techniques, which are valuable skills in the field of network administration and cybersecurity.Also executed several hping3 commands and performed DOS attack using hping3

Lab Assignment 10

Aim: To study and configure Firewalls using IP tables

LO Attainment **: LO6**

Firewall:

A firewall is a system designed to prevent unauthorized access to or from a private network. You can implement a firewall in either hardware or software form, or a combination of both. Generally the firewall has two network interfaces: one for the external side of the network, one for the internal side. Its purpose is to control what traffic is allowed to traverse from one side to the other. As the most basic level, firewalls can block traffic intended for particular IP addresses or server ports.

TCP network traffic moves around a network in packets, which are containers that consist of a packet header—this contains control information such as source and destination addresses, and packet sequence information—and the data (also known as a payload). While the control information in each packet helps to ensure that its associated data gets delivered properly, the elements it contains also provides firewalls a variety of ways to match packets against firewall rules.

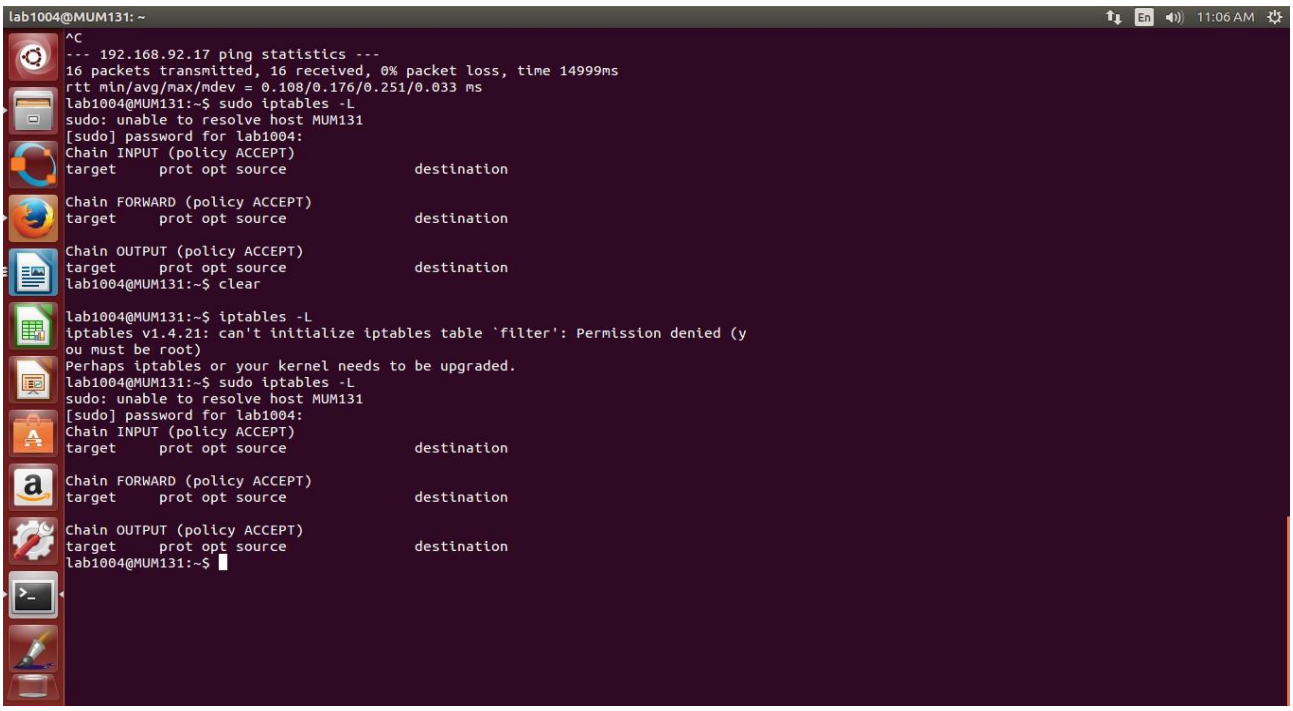
Types of Firewalls

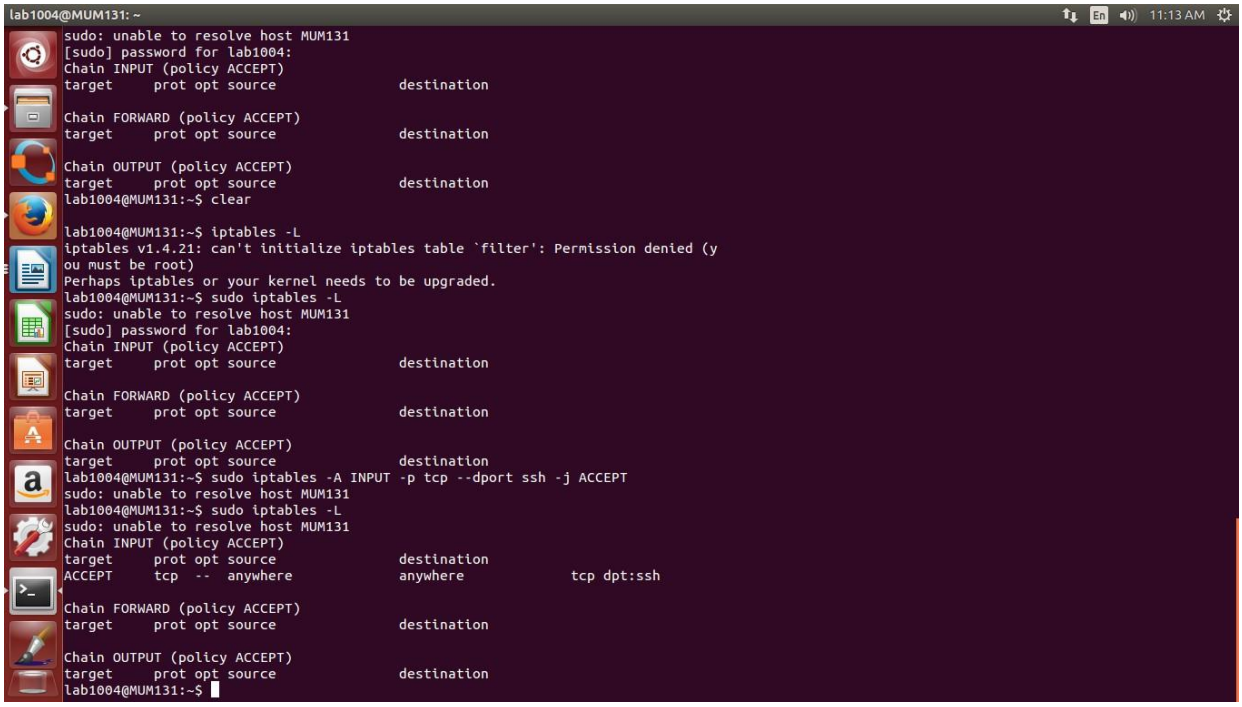
Three basic types of network firewalls: packet filtering (stateless), stateful, and application layer.

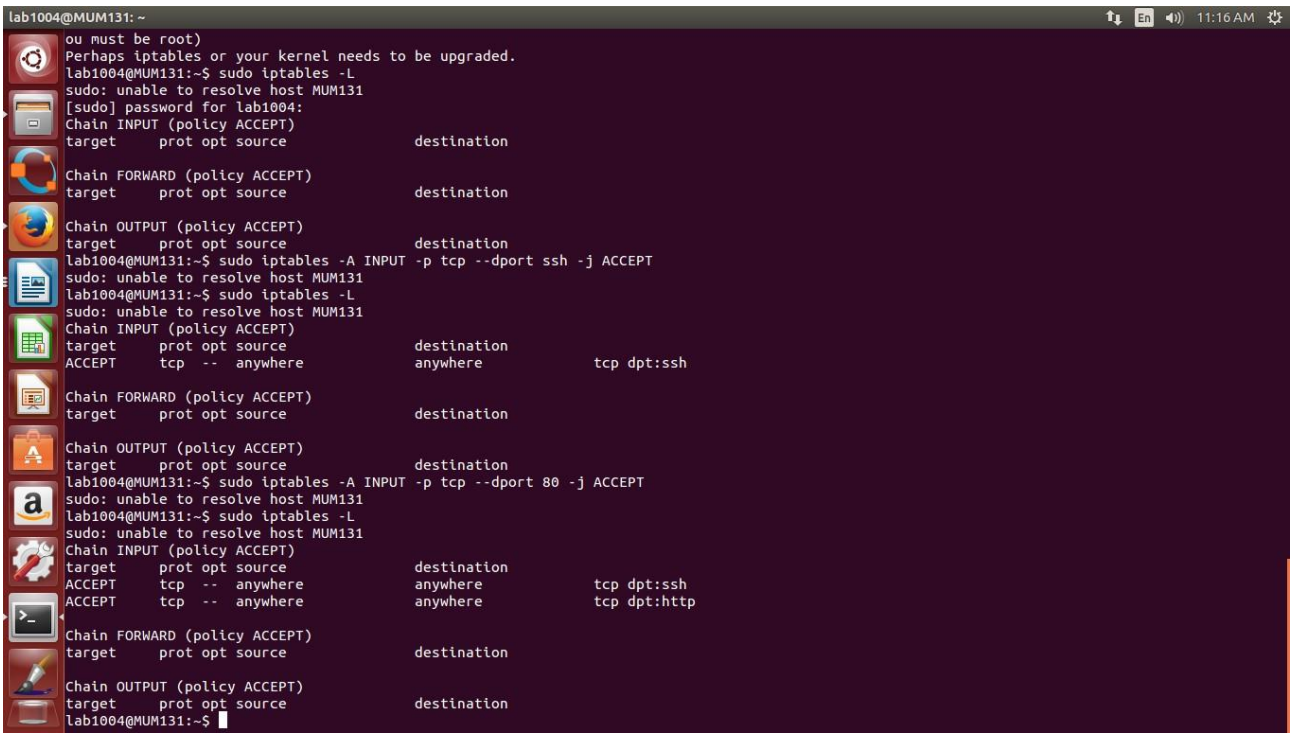
Packet filtering, or stateless, firewalls work by inspecting individual packets in isolation. As such, they are unaware of connection state and can only allow or deny packets based on individual packet headers.

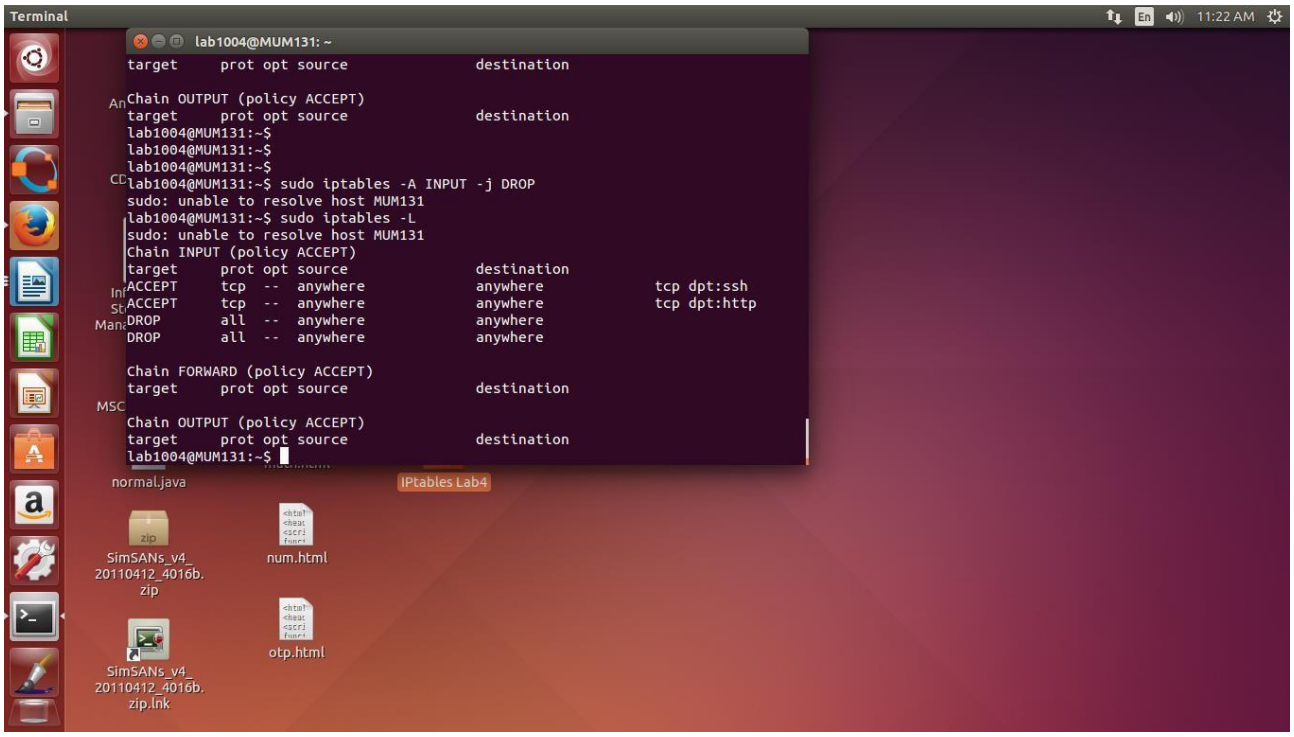
Stateful firewalls are able to determine the connection state of packets, which makes them much more flexible than stateless firewalls. They work by collecting related packets until the connection state can be determined before any firewall rules are applied to the traffic.

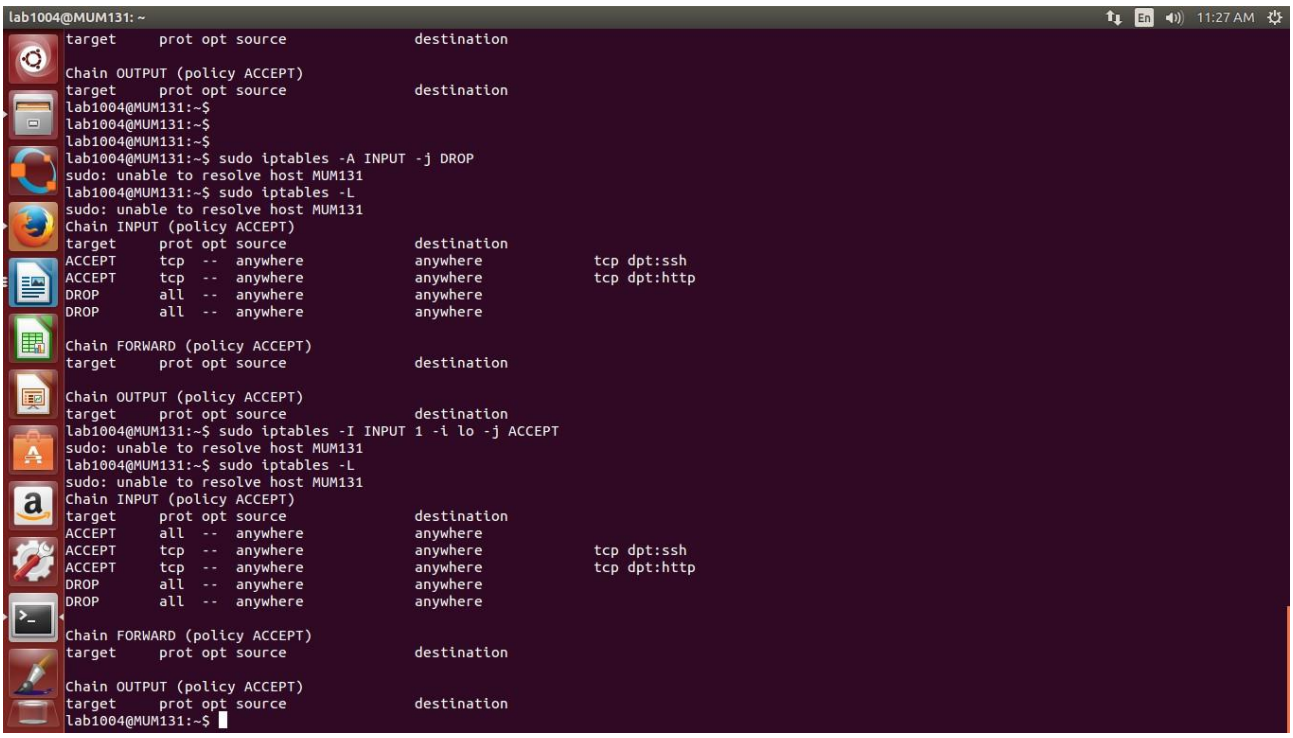
Application firewalls go one step further by analyzing the data being transmitted, which allows network traffic to be matched against firewall rules that are specific to individual services or applications. These are also known as proxy-based firewalls.

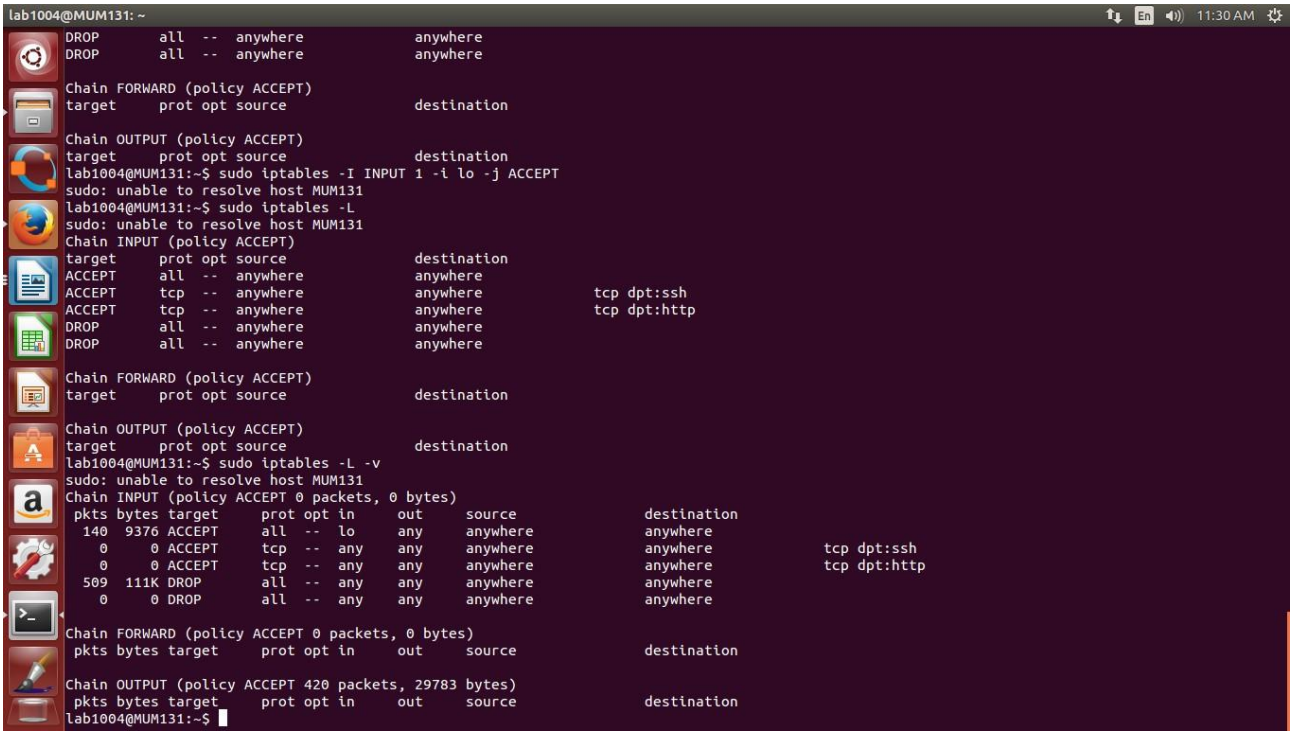

lab1004@MUM131: ~
11:06 AM
^C
--- 192.168.92.17 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 14999ms
rtt min/avg/max/mdev = 0.108/0.176/0.251/0.033 ms
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
[sudo] password for lab1004:
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ clear
lab1004@MUM131:~$ iptables -L
iptables v1.4.21: can't initialize iptables table `filter': Permission denied (y
ou must be root)
Perhaps iptables or your kernel needs to be upgraded.
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
[sudo] password for lab1004:
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$

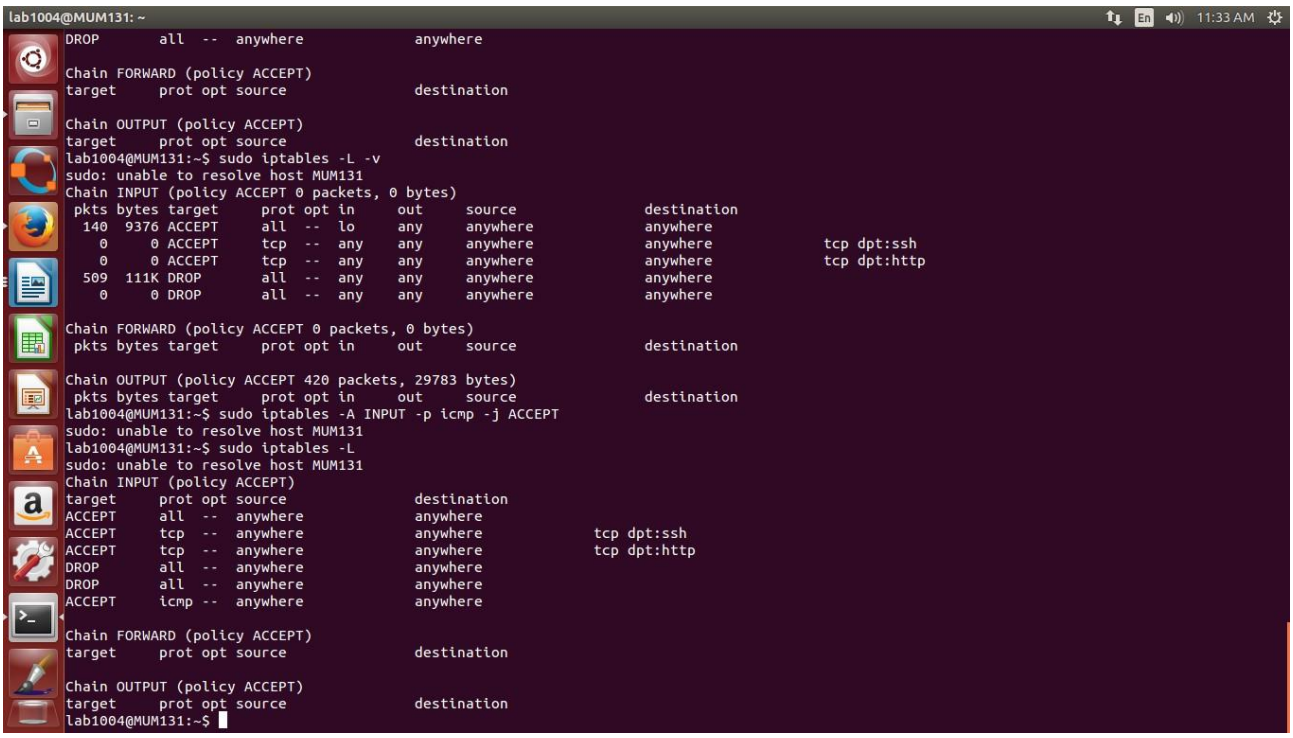

lab1004@MUM131: ~
11:13 AM
sudo: unable to resolve host MUM131
[sudo] password for lab1004:
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ clear
lab1004@MUM131:~$ iptables -L
iptables v1.4.21: can't initialize iptables table `filter': Permission denied (y
ou must be root)
Perhaps iptables or your kernel needs to be upgraded.
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
[sudo] password for lab1004:
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$

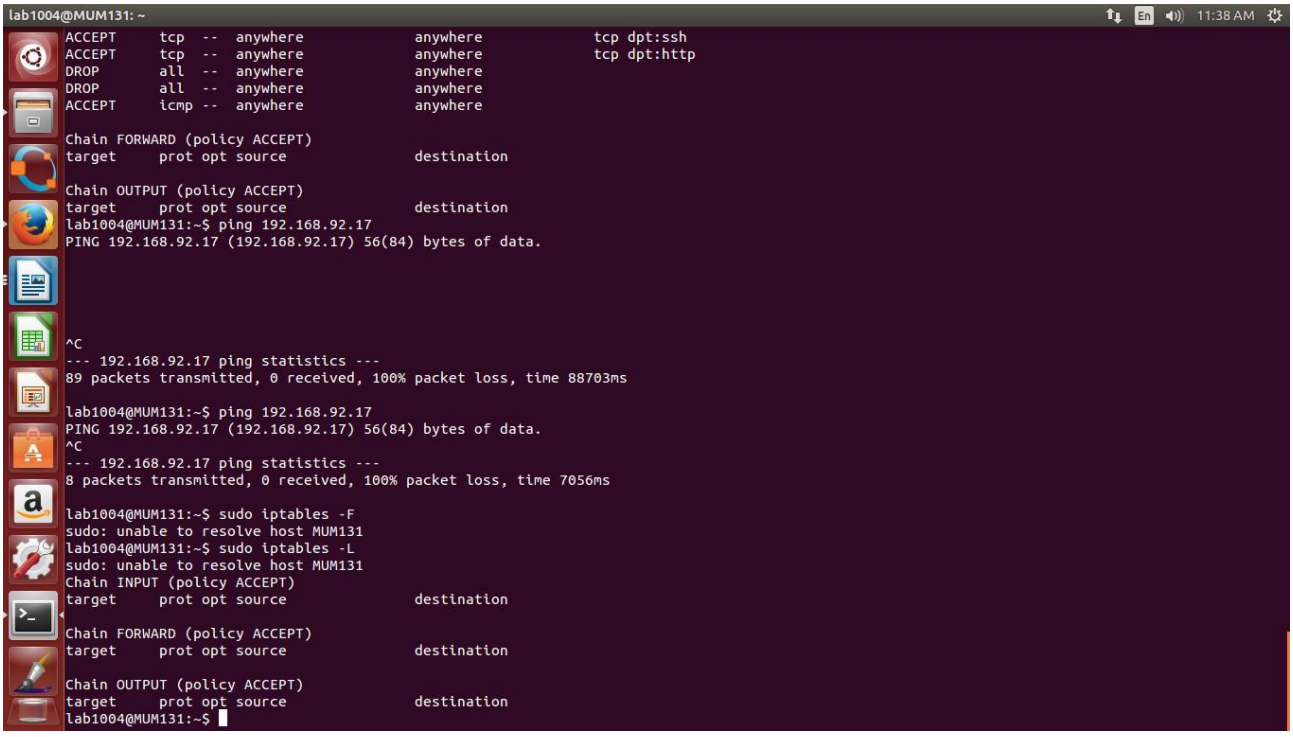

lab1004@MUM131: ~
11:16 AM
ou must be root)
Perhaps iptables or your kernel needs to be upgraded.
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
[sudo] password for lab1004:
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$

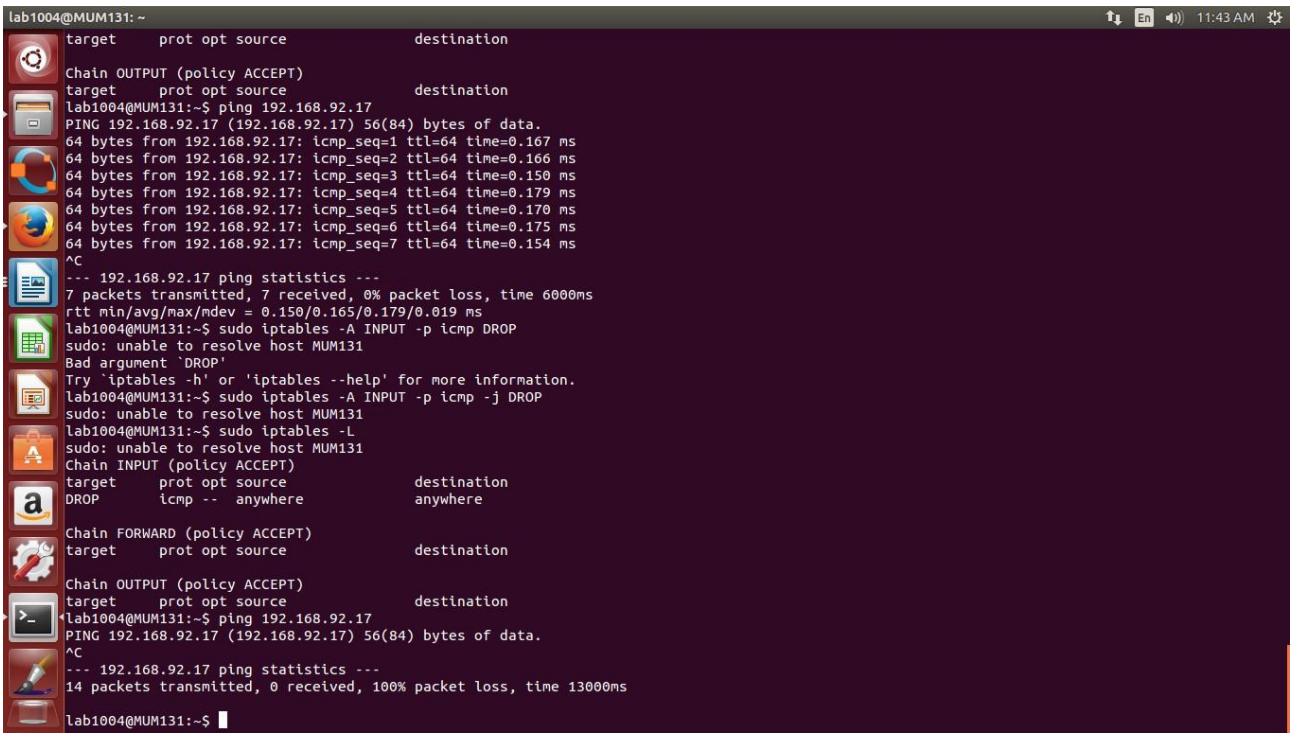

Terminal
11:22 AM
lab1004@MUM131: ~
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$
lab1004@MUM131:~$
lab1004@MUM131:~$
lab1004@MUM131:~$ sudo iptables -A INPUT -j DROP
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$
normal.java
IPtables Lab4
SimSANs_v4_20110412_4016b.zip
num.html
SimSANs_v4_20110412_4016b.zip.lnk
otp.html

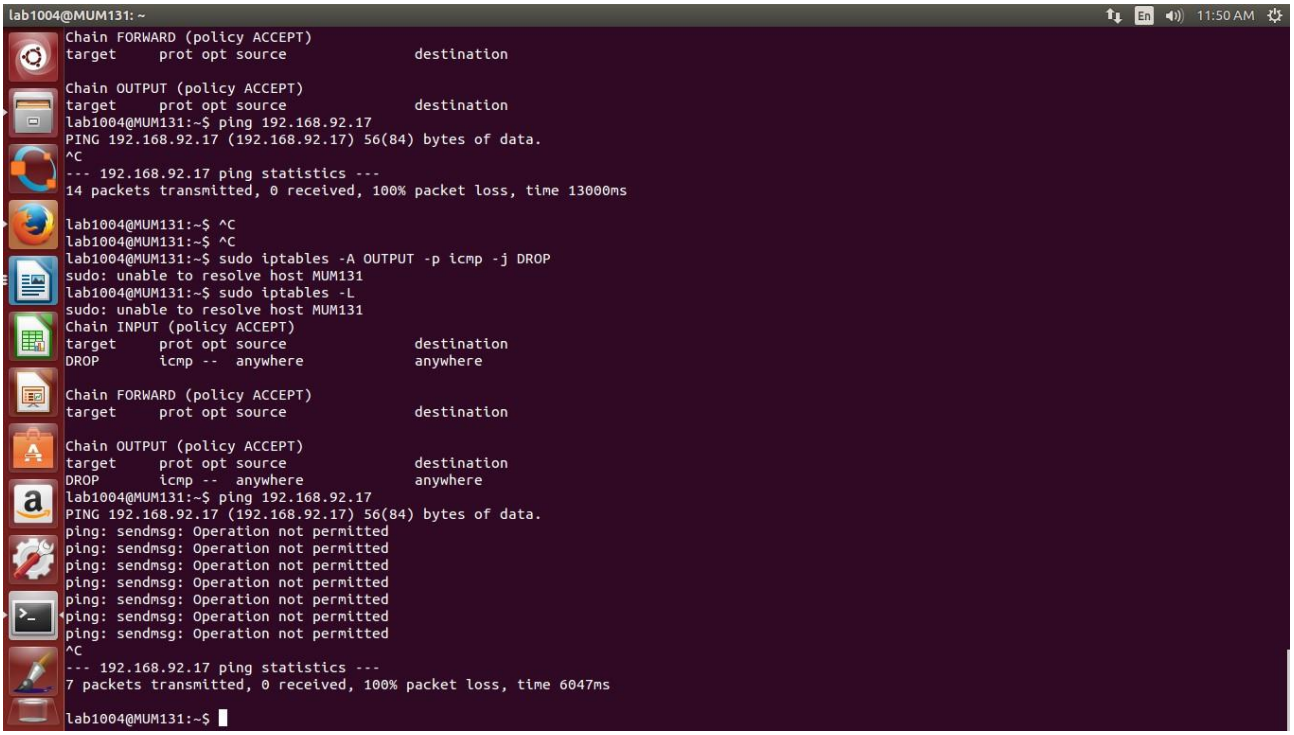

lab1004@MUM131: ~
11:27 AM
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$
lab1004@MUM131:~$
lab1004@MUM131:~$
lab1004@MUM131:~$ sudo iptables -A INPUT -j DROP
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -I INPUT 1 -i lo -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT all -- anywhere anywhere
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$

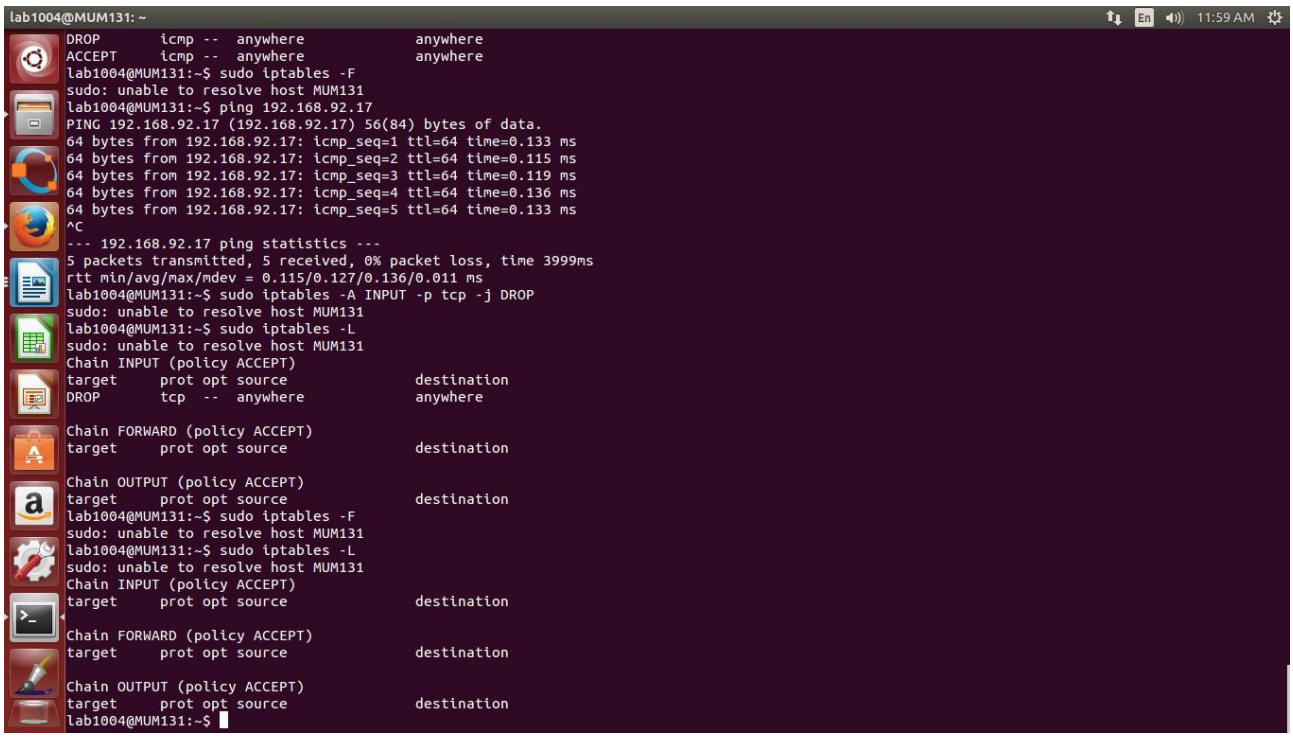

lab1004@MUM131: ~
11:30 AM
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -I INPUT 1 -i lo -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT all -- anywhere anywhere
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -L -v
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
140 9376 ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:ssh
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:http
509 111K DROP all -- any any anywhere anywhere
0 0 DROP all -- any any anywhere anywhere
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 420 packets, 29783 bytes)
pkts bytes target prot opt in out source destination
lab1004@MUM131:~$


lab1004@MUM131: ~
11:33 AM
DROP all -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ sudo iptables -L -v
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
140 9376 ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:ssh
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:http
509 111K DROP all -- any any anywhere anywhere
0 0 DROP all -- any any anywhere anywhere
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 420 packets, 29783 bytes)
pkts bytes target prot opt in out source destination
lab1004@MUM131:~$ sudo iptables -A INPUT -p icmp -j ACCEPT
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT all -- anywhere anywhere
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
ACCEPT icmp -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$


lab1004@MUM131: ~
11:38 AM
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
DROP all -- anywhere anywhere
ACCEPT icmp -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
89 packets transmitted, 0 received, 100% packet loss, time 88703ms
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7056ms
lab1004@MUM131:~$ sudo iptables -F
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$


lab1004@MUM131: ~
11:43 AM
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
64 bytes from 192.168.92.17: icmp_seq=1 ttl=64 time=0.167 ms
64 bytes from 192.168.92.17: icmp_seq=2 ttl=64 time=0.166 ms
64 bytes from 192.168.92.17: icmp_seq=3 ttl=64 time=0.150 ms
64 bytes from 192.168.92.17: icmp_seq=4 ttl=64 time=0.179 ms
64 bytes from 192.168.92.17: icmp_seq=5 ttl=64 time=0.170 ms
64 bytes from 192.168.92.17: icmp_seq=6 ttl=64 time=0.175 ms
64 bytes from 192.168.92.17: icmp_seq=7 ttl=64 time=0.154 ms
^C
--- 192.168.92.17 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 0.150/0.165/0.179/0.019 ms
lab1004@MUM131:~$ sudo iptables -A INPUT -p icmp DROP
sudo: unable to resolve host MUM131
Bad argument `DROP'
Try `iptables -h' or 'iptables --help' for more information.
lab1004@MUM131:~$ sudo iptables -A INPUT -p icmp -j DROP
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
DROP icmp -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13000ms
lab1004@MUM131:~$


lab1004@MUM131: ~
11:50 AM
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13000ms
lab1004@MUM131:~$ ^C
lab1004@MUM131:~$ ^C
lab1004@MUM131:~$ sudo iptables -A OUTPUT -p icmp -j DROP
sudo: unable to resolve host MUM131
lab1004@MUM131:~$ sudo iptables -L
sudo: unable to resolve host MUM131
Chain INPUT (policy ACCEPT)
target prot opt source destination
DROP icmp -- anywhere anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
DROP icmp -- anywhere anywhere
lab1004@MUM131:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 192.168.92.17 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6047ms
lab1004@MUM131:~$



Conclusion :-

We have successfully learned and implemented the concept of firewalls, we learned to see content of iptables ,get more details of the table, append new rules for packet filteration, droping and blocking the packets of specific protocol, etc.

Experiment No 11

Aim : Installing Snort, configuring in Intrusion Detection Mode and writing rules for detecting piging activity .

Lab Outcome :

LO6 : Demonstrate the network security system using open source tools.

Theory :

1. What is Intrusion Detection System?

An Intrusion Detection System (IDS) is a critical cybersecurity tool designed to monitor network traffic, system activities, and user behavior to detect and respond to unauthorized or malicious activities. It serves as a proactive defense mechanism against cyber threats by identifying anomalies, patterns, and signs of potential attacks within a computer network or system. IDS operates by analyzing data in real-time, comparing it against predefined signatures, behavioral baselines, or anomaly detection algorithms.

IDS can be categorized into two main types: Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS monitors network traffic at various points within the network, identifying signs of intrusion or malicious activity that might bypass perimeter defenses. On the other hand, HIDS focuses on individual hosts or endpoints, analyzing system logs, files, and activities for any signs of compromise.

The primary goal of an IDS is to provide early detection of cyber threats, mitigate the risk of security breaches, and enable rapid response to incidents. By generating alerts or alarms when suspicious behavior is identified, IDS empowers network administrators and security teams to take appropriate action and protect the integrity, confidentiality, and availability of critical systems and data.

2. What are different modes in which Snort works? (refer user mannual on snort.org for this)?

Snort, a widely used open-source Intrusion Detection System (IDS), operates in several distinct modes, each catering to specific monitoring and analysis needs. As outlined in the official Snort user manual available on snort.org, these modes offer flexibility and versatility in network security:

1. Sniffer Mode: In this mode, Snort behaves like a packet sniffer, capturing and displaying network traffic in real-time. It is valuable for troubleshooting network issues and gaining insights into the flow of data. However, it doesn't involve analysis or rule-based detection.

2. Packet Logger Mode: In this mode, Snort logs captured packets to disk for later analysis. It's useful for preserving evidence and performing forensic investigations after potential security incidents.

3. Network Intrusion Detection Mode: This is the primary and most crucial mode of Snort. In this mode, Snort analyzes network traffic against a set of predefined rules and signatures. If it detects any patterns or behaviors that match the specified rules, it generates alerts to notify administrators of potential security threats. This mode enables real-time detection and response to unauthorized or malicious activities.

4. Network Intrusion Prevention Mode: This advanced mode not only detects but also actively prevents identified attacks by blocking or mitigating suspicious traffic. It involves inline deployment and requires careful configuration to avoid disrupting legitimate network communication.

Each of these modes addresses different use cases and security requirements, making Snort a versatile tool that can adapt to various monitoring and protection needs. By offering multiple modes of operation, Snort empowers security professionals to choose the mode that aligns best with their specific goals and helps enhance the overall security posture of their networks.

3. Write the commands used for installing snort, editing its configuration file and configurig it in intrusion detection mode ?

Here are the commands you need to follow for installing Snort, editing its configuration file, and configuring it in intrusion detection mode as per the instructions provided:

1. Check the interface name:
```shell
ifconfig
```

2. Install Snort:
```shell
sudo apt-get update
sudo apt-get install snort
```

3. During installation, specify the interface name observed in step 1 when asked.

4. Edit Snort configuration file:
```shell
sudo gedit /etc/snort/snort.conf
```

5. Make the following changes in the configuration file:
   - In section 1, set:
     ```
     ipvar HOME_NET 192.168.44.0/24
     ```

```
6. Open a new terminal and open the `ftp.rules` file (optional):
```shell
sudo gedit /etc/snort/rules/ftp.rules
```

7. In a new terminal, validate rules:
```shell
sudo snort -T -c /etc/snort/snort.conf -i ens33
```

8. Start Snort in NIDS mode:
```shell
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

9. On Kali Linux, run port scanning:
```shell
nmap 192.168.44.128
```

10. Observe the output in the Snort terminal.

11. Ping the Ubuntu machine from Kali Linux:
```shell
ping 192.168.44.128
```

12. Adding a rule for detecting ping activity:
  a. Create a local.rules file:
  ```shell
  sudo gedit /etc/snort/rules/local.rules
  ```
  b. Write the rule in local.rules:
  ```
  alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001;
rev:001; classtype:icmp-event;)
  ```
  c. Save the local.rules file.

  d. Comment the following lines in snort.conf:
  ```
  include $RULE_PATH/icmp.rules
  include $RULE_PATH/icmp-info.rules
  ```
  e. Include the local.rules file in the configuration:
  ```
  include $RULE_PATH/local.rules

```

f. Validate changes in snort.conf:
```shell
sudo snort -T -c /etc/snort/snort.conf -i ens33
```

g. Start Snort in Intrusion Detection Mode:
```shell
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

h. Ping the Ubuntu machine from Kali and observe alerts.

i. Compare alerts generated using different rules.

Ensure to follow each step carefully and observe the outputs as instructed to effectively configure and use Snort in intrusion detection mode.

Output:

Conclusion:

In conclusion, this assignment involved the installation and configuration of Snort, a powerful Intrusion Detection System. By following the step-by-step instructions, we successfully installed Snort, edited its configuration file, and executed rules to detect ICMP activities. This hands-on experience enhanced our understanding of network security and IDS functionality.
```

Roll Number:- 122
Name:-Moheet Shendarkar
Date:- 13/09/2023

Lab Assignment No:-13

Aim:- Explore the GPG tool of Linux to implement email security

Lab Outcome Attained :- LO6

Theory:-

What is private key ring and public key ring ?

a)Public key ring

The public key ring contains the public keys of other users. These keys are made available to the public so that anyone can encrypt messages to the user. The public key ring is typically shared with other users by exporting it to a file or by adding it to a PGP keyserver.

The public key contains the following information:

The user's name or email address
The user's fingerprint, which is a unique identifier for the key
The key's algorithm and strength
The key's expiration date
When someone wants to encrypt a message to you, they will use your public key. The message will be encrypted using the public key, but it can only be decrypted using the corresponding private key.

b)Private key ring

The private key ring contains the private keys of the user. These keys are kept secret and should not be shared with anyone. The private key ring is typically protected by a password or passphrase.

The private key contains the following information:

The user's name or email address
The user's fingerprint, which is a unique identifier for the key
The key's algorithm and strength
The key's expiration date
The private key is used to decrypt messages that have been encrypted with the user's public key. It is also used to sign messages, which allows the recipient to verify that the message was sent by the intended sender.

The public key ring and the private key ring are essential for using PGP. They allow users to encrypt and decrypt messages securely.

Write the commands used for key generation, export and import of keys and signing and encrypting the message in gpg tool.

Key generation

The following command generates a new GPG key pair:

gpg --gen-key
This command will prompt you for some information, such as your name, email address, and key length.

Export and import of keys

The following command exports the public key to a file:

gpg --export --output public.key
The following command imports the public key from a file:

gpg --import public.key
The following command exports the private key to a file:

gpg --export-secret-key --output private.key
The following command imports the private key from a file:

gpg --import-secret-key private.key
Signing and encrypting the message

The following command signs a message:

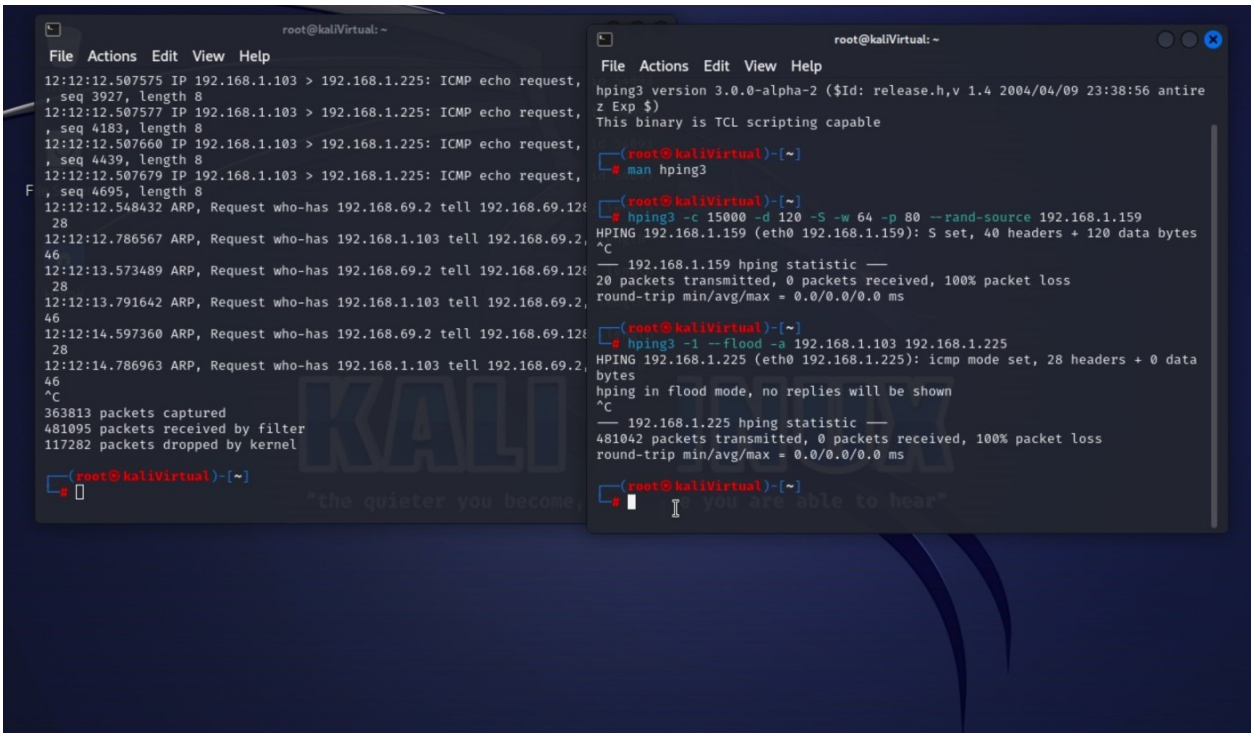
gpg --sign message.txt
The following command encrypts a message:

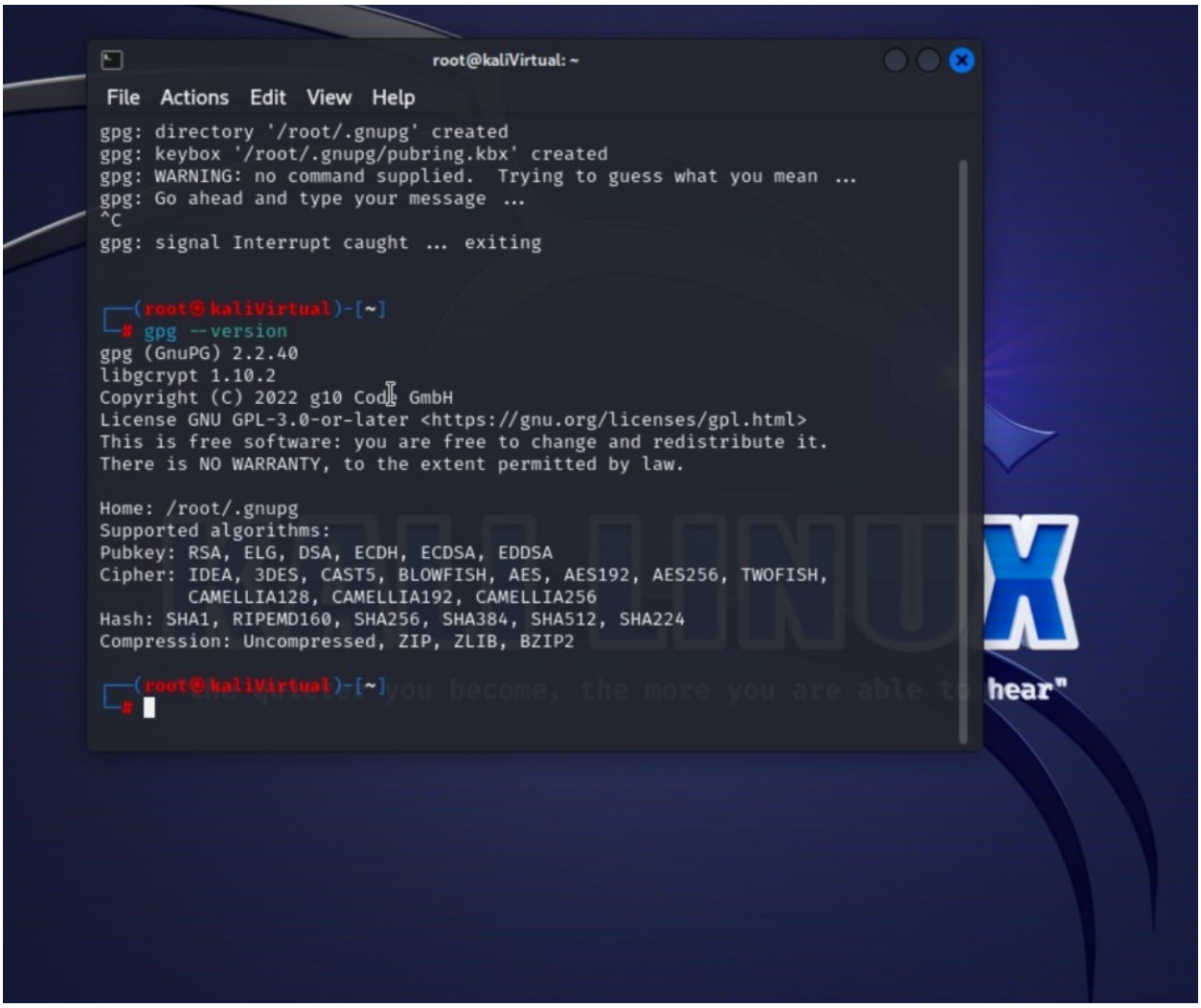
gpg --encrypt --recipient recipient@example.com message.txt
The recipient can then decrypt the message using their private key.

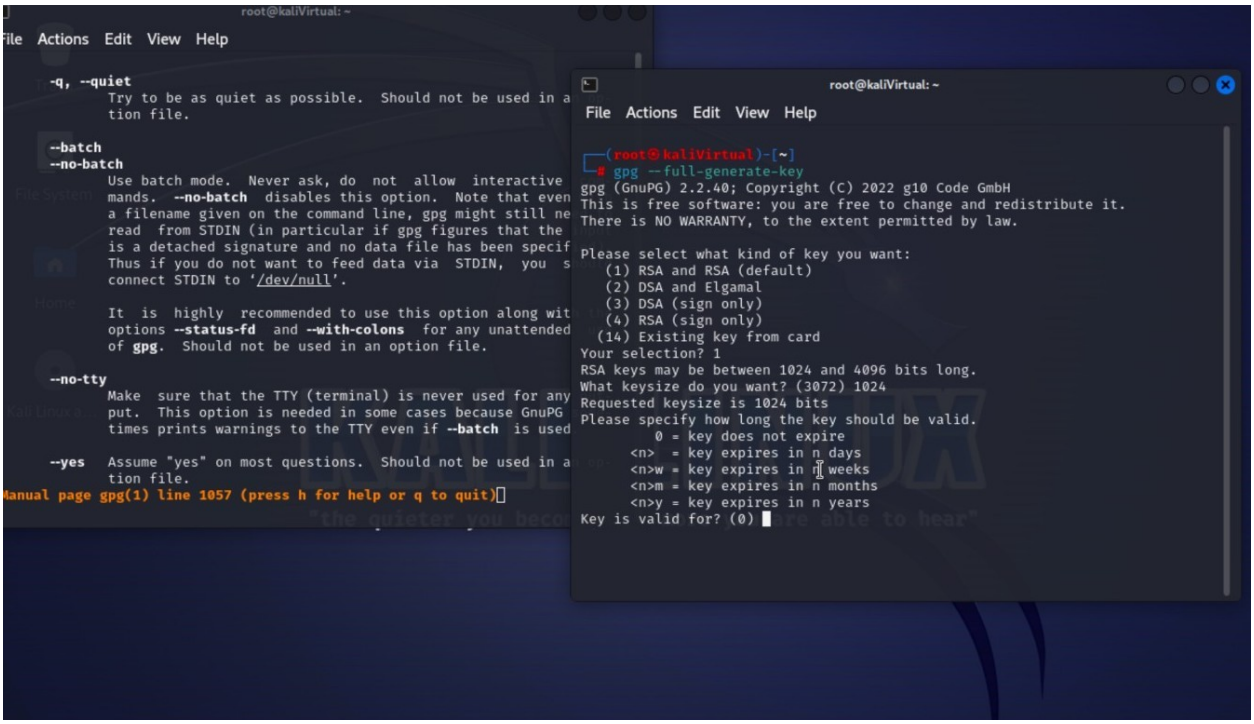
Some additional details about the commands:

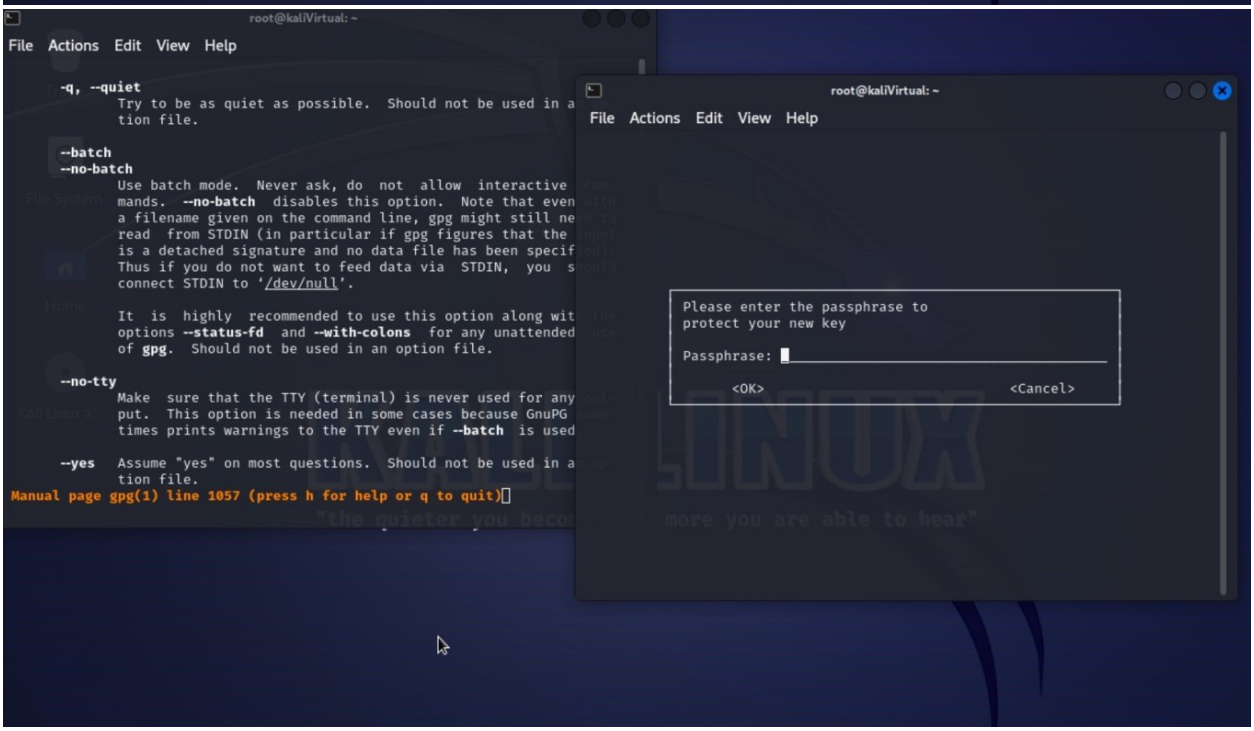
The gpg command is the main GPG command.
The **--gen-key** option generates a new GPG key pair.
The **--export** option exports a key to a file.
The **--import** option imports a key from a file.
The **--sign** option signs a message.
The **--encrypt** option encrypts a message.
The **--recipient** option specifies the recipient of the encrypted message.

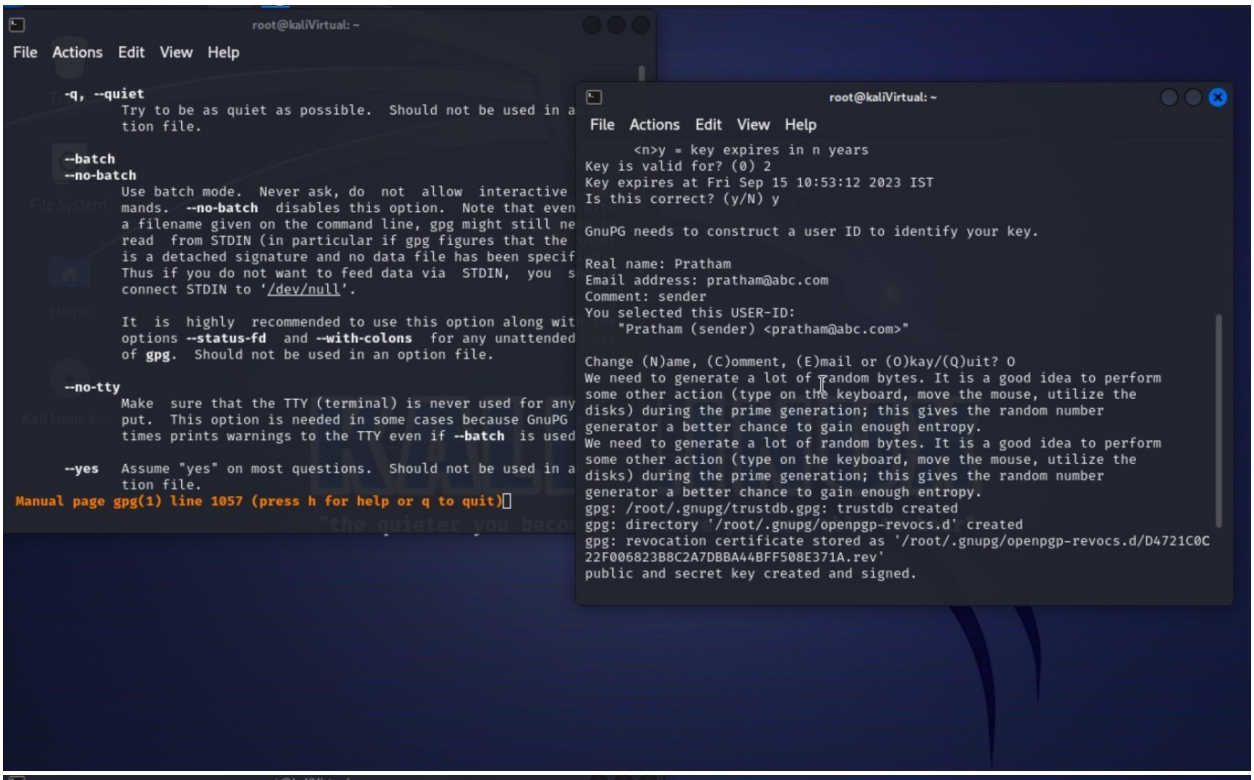
Output Screenshots

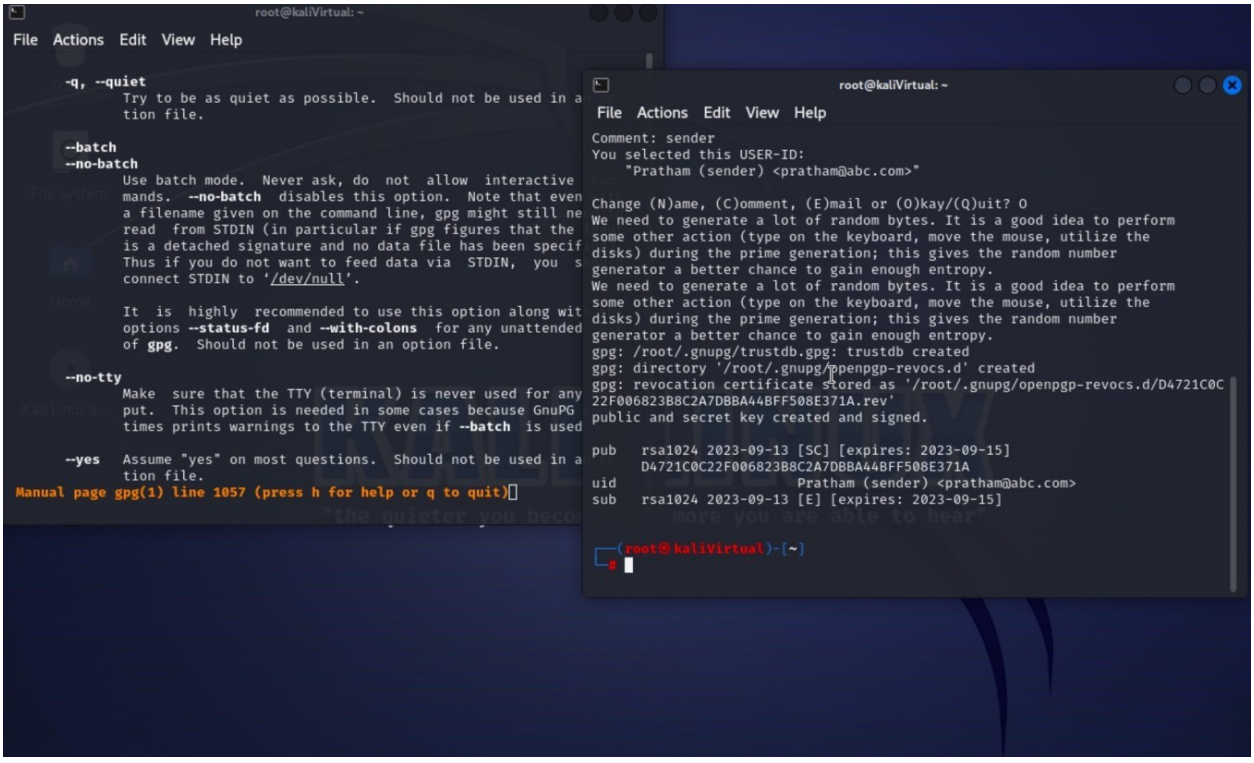

root@kaliVirtual: ~
File Actions Edit View Help
12:12:12.507575 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,
, seq 3927, length 8
12:12:12.507577 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,
, seq 4183, length 8
12:12:12.507660 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,
, seq 4439, length 8
12:12:12.507679 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,
, seq 4695, length 8
12:12:12.548432 ARP, Request who-has 192.168.69.2 tell 192.168.69.128
28
12:12:12.786567 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,
46
12:12:13.573489 ARP, Request who-has 192.168.69.2 tell 192.168.69.128
28
12:12:13.791642 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,
46
12:12:14.597360 ARP, Request who-has 192.168.69.2 tell 192.168.69.128
28
12:12:14.786963 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,
46
^C
363813 packets captured
481095 packets received by filter
117282 packets dropped by kernel
(root㉿kaliVirtual)-[~]
#
root@kaliVirtual: ~
File Actions Edit View Help
hping3 version 3.0.0-alpha-2 ($Id: release.h,v 1.4 2004/04/09 23:38:56 antire
z Exp $)
This binary is TCL scripting capable
(root㉿kaliVirtual)-[~]
man hping3
(root㉿kaliVirtual)-[~]
hping3 -c 15000 -d 120 -S -w 64 -p 80 --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): S set, 40 headers + 120 data bytes
^C
— 192.168.1.159 hping statistic —
20 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
(root㉿kaliVirtual)-[~]
hping3 -1 --flood -a 192.168.1.103 192.168.1.225
HPING 192.168.1.225 (eth0 192.168.1.225): icmp mode set, 28 headers + 0 data
bytes
hping in flood mode, no replies will be shown
^C
— 192.168.1.225 hping statistic —
481042 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
(root㉿kaliVirtual)-[~]
#

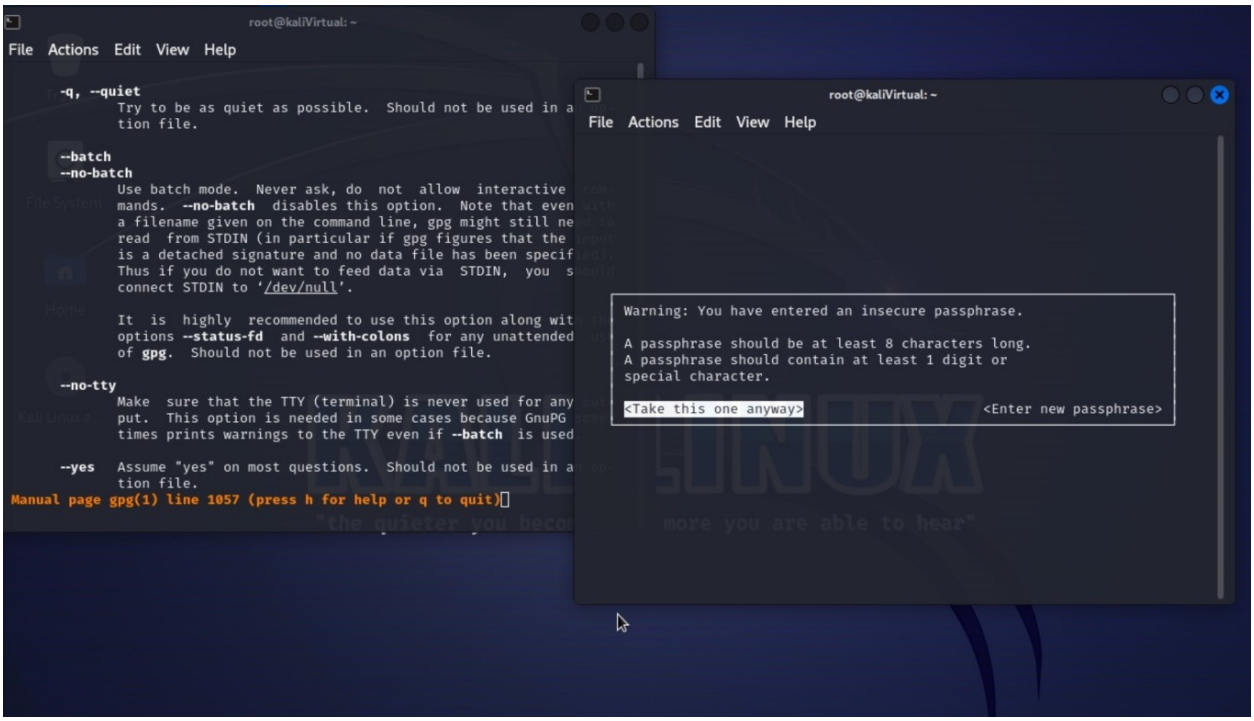

root@kaliVirtual: ~
File Actions Edit View Help
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: WARNING: no command supplied. Trying to guess what you mean ...
gpg: Go ahead and type your message ...
^C
gpg: signal Interrupt caught ... exiting
(root㉿kaliVirtual)-[~]
gpg --version
gpg (GnuPG) 2.2.40
libgcrypt 1.10.2
Copyright (C) 2022 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
(root㉿kaliVirtual)-[~]
#

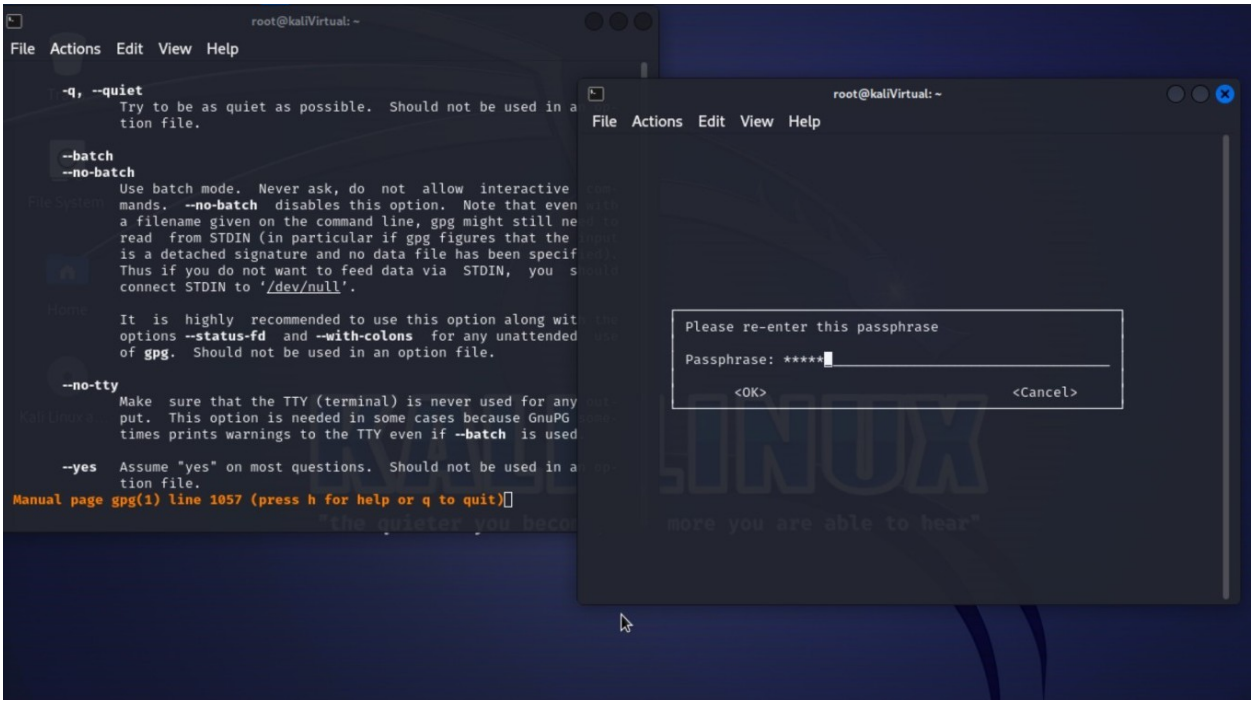

root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
┌──(root㉿kaliVirtual)-[~]
└─# gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0)

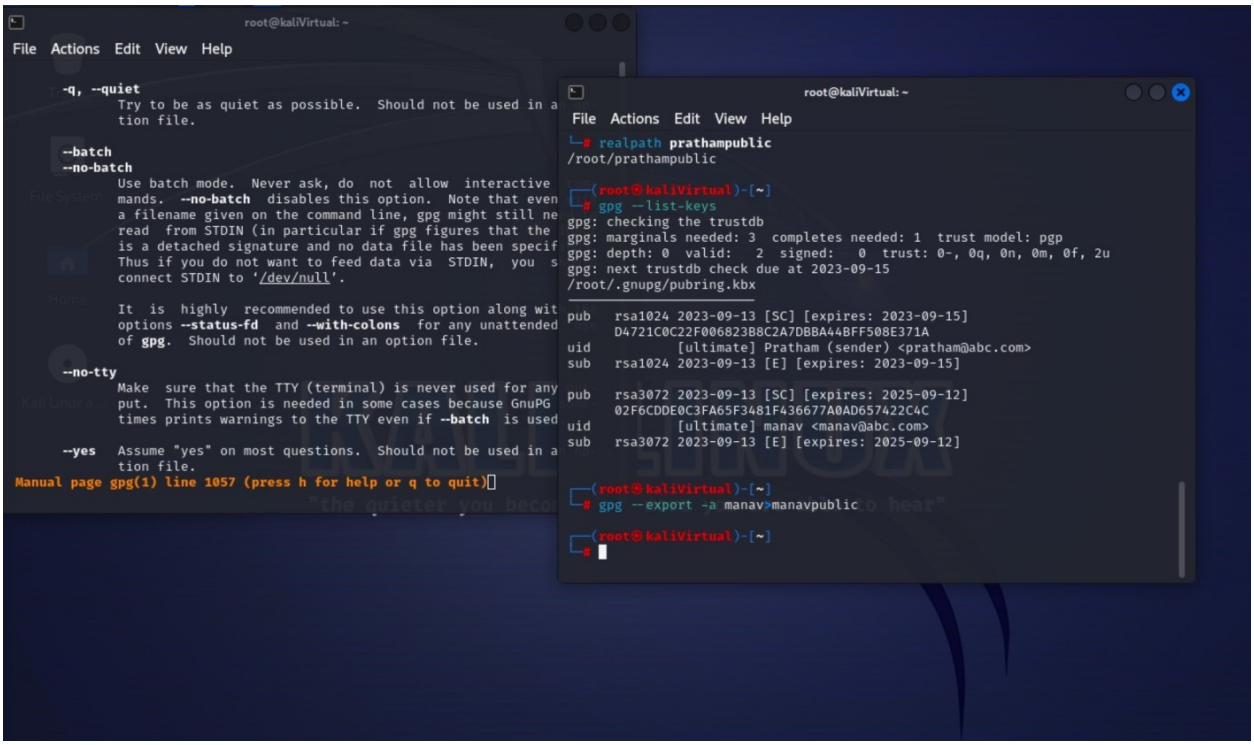

root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
Please enter the passphrase to
protect your new key
Passphrase:
<OK>
<Cancel>

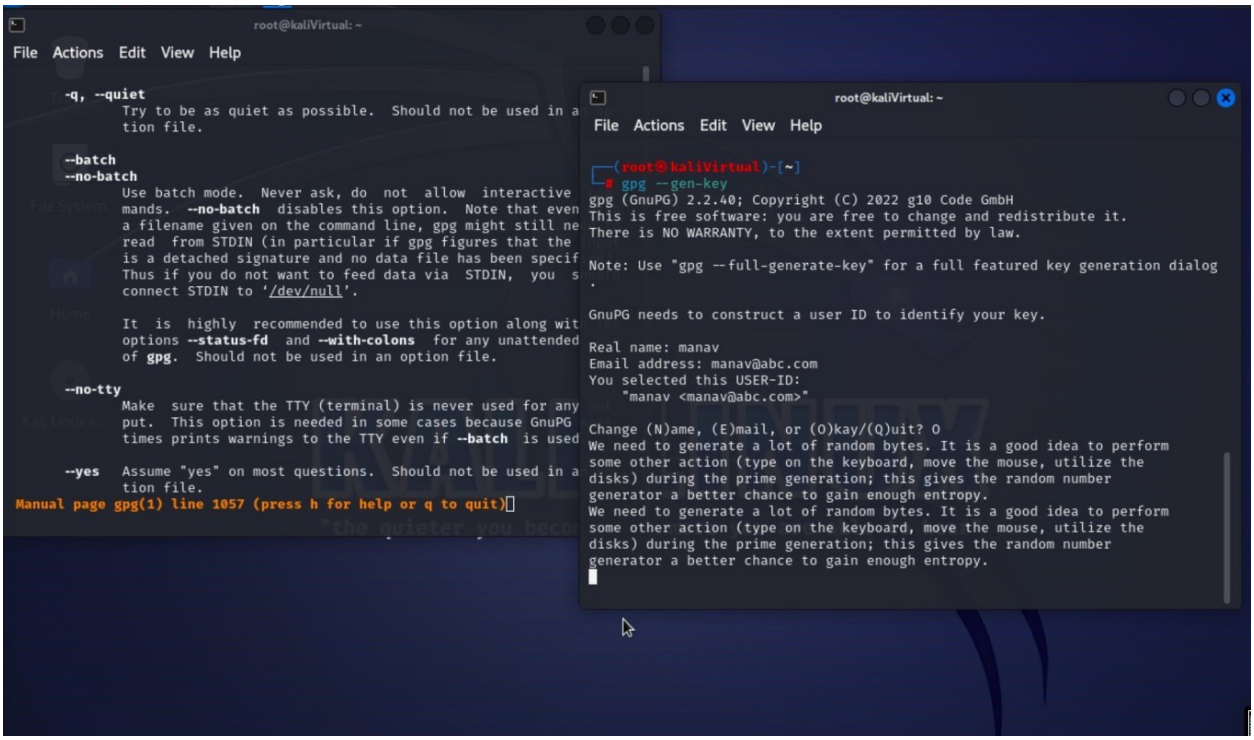

root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
<n>y = key expires in n years
Key is valid for? (0) 2
Key expires at Fri Sep 15 10:53:12 2023 IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: Pratham
Email address: pratham@abc.com
Comment: sender
You selected this USER-ID:
"Pratham (sender) <pratham@abc.com>"
Change (N)ame, (C)omment, (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: /root/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/D4721C0C
22F006823B8C2A7DBBA44BFF508E371A.rev'
public and secret key created and signed.

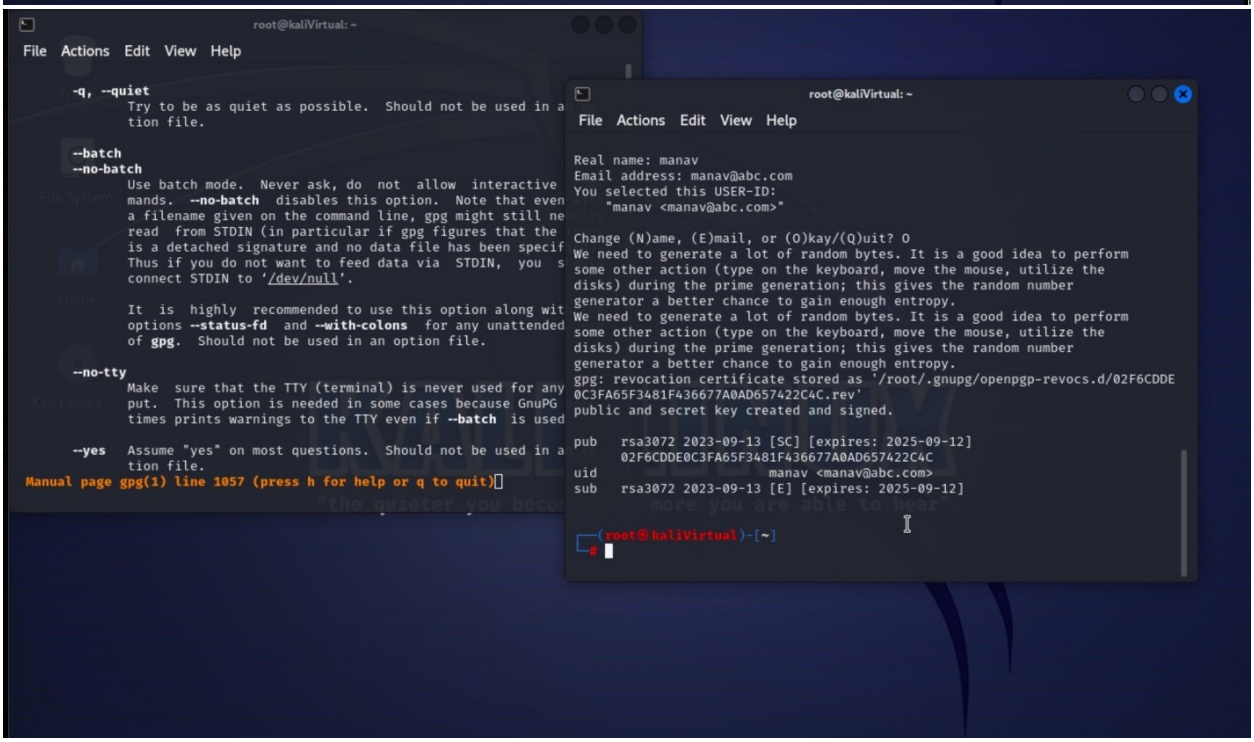

root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
Comment: sender
You selected this USER-ID:
"Pratham (sender) <pratham@abc.com>"
Change (N)ame, (C)omment, (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: /root/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/D4721C0C
22F006823B8C2A7DBBA44BFF508E371A.rev'
public and secret key created and signed.
pub rsa1024 2023-09-13 [SC] [expires: 2023-09-15]
D4721C0C22F006823B8C2A7DBBA44BFF508E371A
uid Pratham (sender) <pratham@abc.com>
sub rsa1024 2023-09-13 [E] [expires: 2023-09-15]
(root@kaliVirtual)-[~]
#


root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
Warning: You have entered an insecure passphrase.
A passphrase should be at least 8 characters long.
A passphrase should contain at least 1 digit or
special character.
<Take this one anyway>
<Enter new passphrase>


root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
Please re-enter this passphrase
Passphrase: *****
<OK>
<Cancel>


root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
└─# realpath prathampublic
/root/prathampublic
┌──(root㉿kaliVirtual)-[~]
└─# gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2023-09-15
/root/.gnupg/pubring.kbx
pub rsa1024 2023-09-13 [SC] [expires: 2023-09-15]
D4721C0C22F006823B8C2A7DBBA44BFF508E371A
uid [ultimate] Pratham (sender) <pratham@abc.com>
sub rsa1024 2023-09-13 [E] [expires: 2023-09-15]
pub rsa3072 2023-09-13 [SC] [expires: 2025-09-12]
02F6CDDE0C3FA65F3481F436677A0AD657422C4C
uid [ultimate] manav <manav@abc.com>
sub rsa3072 2023-09-13 [E] [expires: 2025-09-12]
┌──(root㉿kaliVirtual)-[~]
└─# gpg --export -a manav>manavpublic
┌──(root㉿kaliVirtual)-[~]
└─#


root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
(root㉿kaliVirtual)-[~]
gpg --gen-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
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: manav
Email address: manav@abc.com
You selected this USER-ID:
"manav <manav@abc.com>"
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.


root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.
--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.
It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used
--yes Assume "yes" on most questions. Should not be used in a
tion file.
Manual page gpg(1) line 1057 (press h for help or q to quit)
root@kaliVirtual: ~
File Actions Edit View Help
Real name: manav
Email address: manav@abc.com
You selected this USER-ID:
"manav <manav@abc.com>"
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: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/02F6CDDE
0C3FA65F3481F436677A0AD657422C4C.rev'
public and secret key created and signed.
pub rsa3072 2023-09-13 [SC] [expires: 2025-09-12]
02F6CDDE0C3FA65F3481F436677A0AD657422C4C
uid manav <manav@abc.com>
sub rsa3072 2023-09-13 [E] [expires: 2025-09-12]
(root㉿kaliVirtual)-[~]
#

Conclusion:-

Learnt about GPG tool in linux and how it provides email security , executed several commands related to GPG and also explored more about public key ring and private key rings

Experiment No 3

Aim : Block Cipher modes of operations using Advanced Encryption Techniques.

Lab Outcome :

LO2

Theory :

1. AES Algorithm?

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm known for its security and efficiency. AES is a block cipher, which means it operates on fixedsize blocks of data and applies a series of transformations to encrypt or decrypt the data. It was adopted by the U.S. government as a standard encryption algorithm in 2001 and has since become a fundamental component of modern cryptography.

Cipher Type:

AES is a symmetric key cipher, also known as a secret-key or private-key cipher. This means that the same secret key is used for both encryption and decryption. The security of AES relies on the strength of the secret key, making it essential to keep the key secret and protected.

Number of Rounds:

AES operates in multiple rounds of transformations to ensure strong security. The number of rounds varies based on the key size:

- For AES-128: 10 rounds

- For AES-192: 12 rounds - For AES-256: 14 rounds

Key Size:

AES supports three different key sizes: 128 bits, 192 bits, and 256 bits. The key size directly affects the algorithm's security, with larger key sizes generally providing higher levels of security.

Block Size:

AES has a fixed block size of 128 bits (16 bytes). This means that the input plaintext is divided into blocks of 128 bits each for encryption or decryption.

Operations in Each Round:

Each round of AES consists of several cryptographic operations, including SubBytes, ShiftRows, MixColumns, and AddRoundKey. Here's a brief overview of these operations:

1. SubBytes:

In this operation, each byte of the input block is replaced by a corresponding byte from a fixed substitution table called the S-box. The S-box is designed to introduce confusion in the data and provide non-linearity to the encryption process.

2. ShiftRows:

In this step, the rows of the block are shifted by varying numbers of bytes. The first row is not shifted, the second row is shifted by one byte to the left, the third row by two bytes, and the fourth row by three bytes. This operation ensures that the data is spread out in a way that contributes to the diffusion property of encryption.

3. MixColumns:

This step operates on the columns of the block, treating each column as a four-term polynomial. MixColumns uses matrix multiplication operations to mix the bytes within each column. This operation further enhances the encryption's diffusion and confusion properties.

4. AddRoundKey:

A round key is generated from the main encryption key for each round. In the AddRoundKey step, each byte of the block is bitwise XORed with the corresponding byte of the round key. This step ensures that the input data is mixed with the current round's key, providing additional security.

After completing the specified number of rounds, the AES encryption process is complete. Decryption involves applying the inverse of each operation in reverse order using the same round keys.

AES's combination of substitution, permutation, diffusion, and confusion operations, along with the varying number of rounds based on key size, contributes to its robust security and widespread adoption in secure communication, data storage, and various cryptographic applications.

2. With diagram explain in brief block cipher modes of operation:

ECB mode

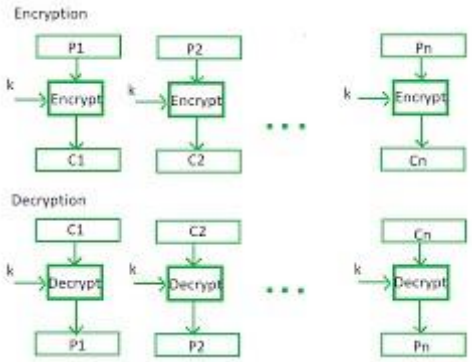
CBC mode

OFB mode

Counter mode

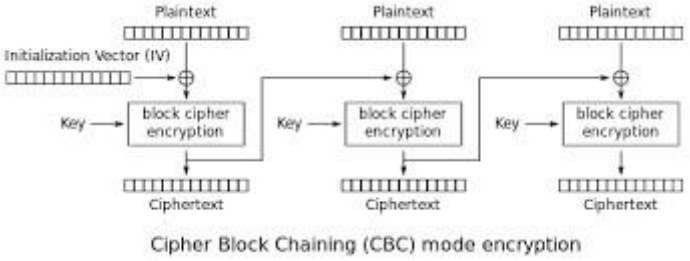
Block cipher modes of operation are techniques used to apply a block cipher, which is a cryptographic algorithm that encrypts fixed-size blocks of data, to larger amounts of data. These modes determine how blocks of plaintext are encrypted and how the resulting ciphertext is generated. Let's explore four common block cipher modes of operation: ECB, CBC, OFB, and Counter mode, along with a brief explanation and diagrams for each.

1. ECB (Electronic Codebook) Mode:
ECB mode is the simplest block cipher mode. It encrypts each block of plaintext independently using the same key, resulting in a corresponding block of ciphertext. While simple, ECB has some weaknesses. Identical plaintext blocks will produce identical ciphertext blocks, which can leak information, and it doesn't provide semantic security.



2. CBC (Cipher Block Chaining) Mode:

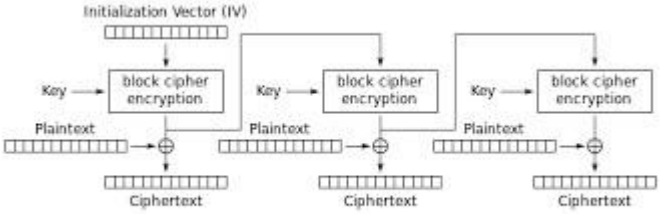
CBC mode addresses the weaknesses of ECB mode by introducing an Initialization Vector (IV) and chaining blocks together. Each plaintext block is XORed with the previous ciphertext block (or the IV for the first block), and then encrypted. This chaining introduces randomness and prevents identical blocks from producing identical ciphertext blocks. CBC is widely used and offers better security.



Cipher Block Chaining (CBC) mode encryption

3. OFB (Output Feedback) Mode:

OFB mode transforms the block cipher into a stream cipher by generating a keystream of random data blocks using the encryption process. This keystream is then XORed with the plaintext to produce the ciphertext. The advantage of OFB is that errors in ciphertext transmission do not propagate, as they would in CBC. However, it doesn't offer integrity checking or error detection.



Output Feedback (OFB) mode encryption

4. Counter Mode:

Counter mode turns a block cipher into a stream cipher by using a counter to generate a sequence of unique values. Each counter value is encrypted with the key to produce a keystream, which is then XORed with the plaintext to create the ciphertext. Counter mode is highly parallelizable and can be more efficient than other modes. It's also suitable for applications like disk encryption and random number generation.

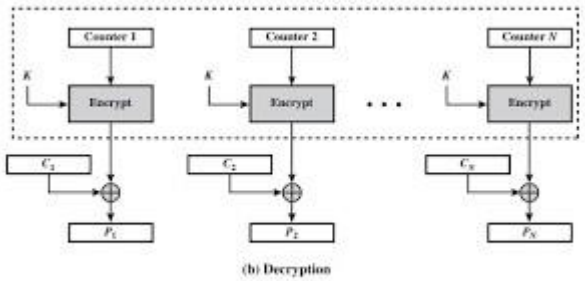
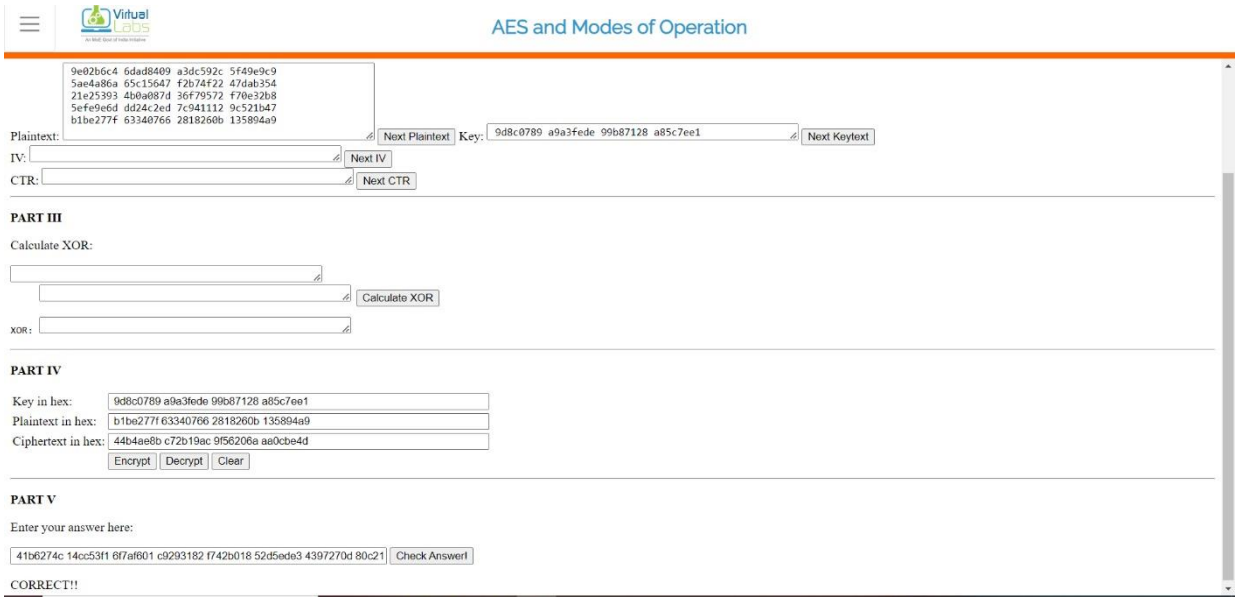
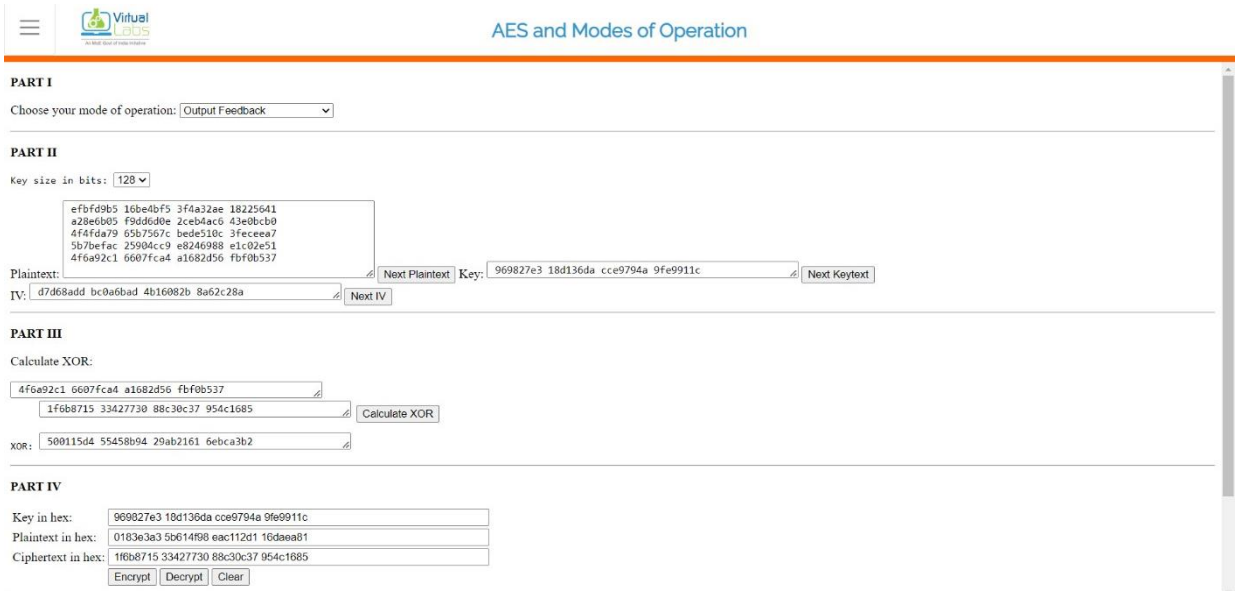


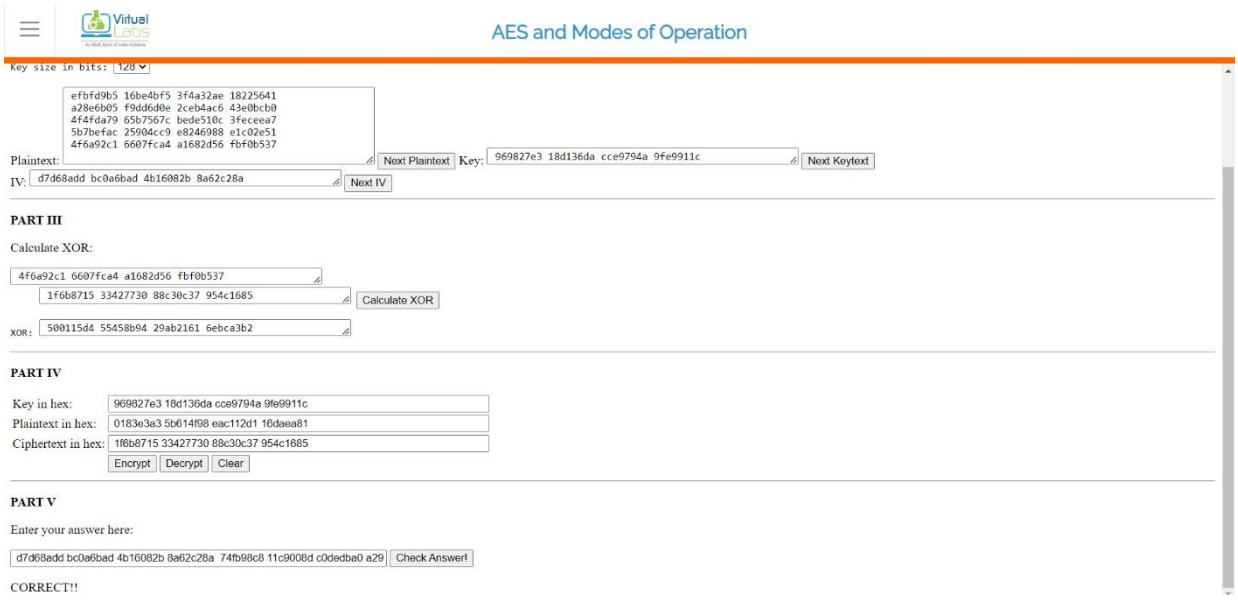
Figure 6.7 Counter (CTR) Mode

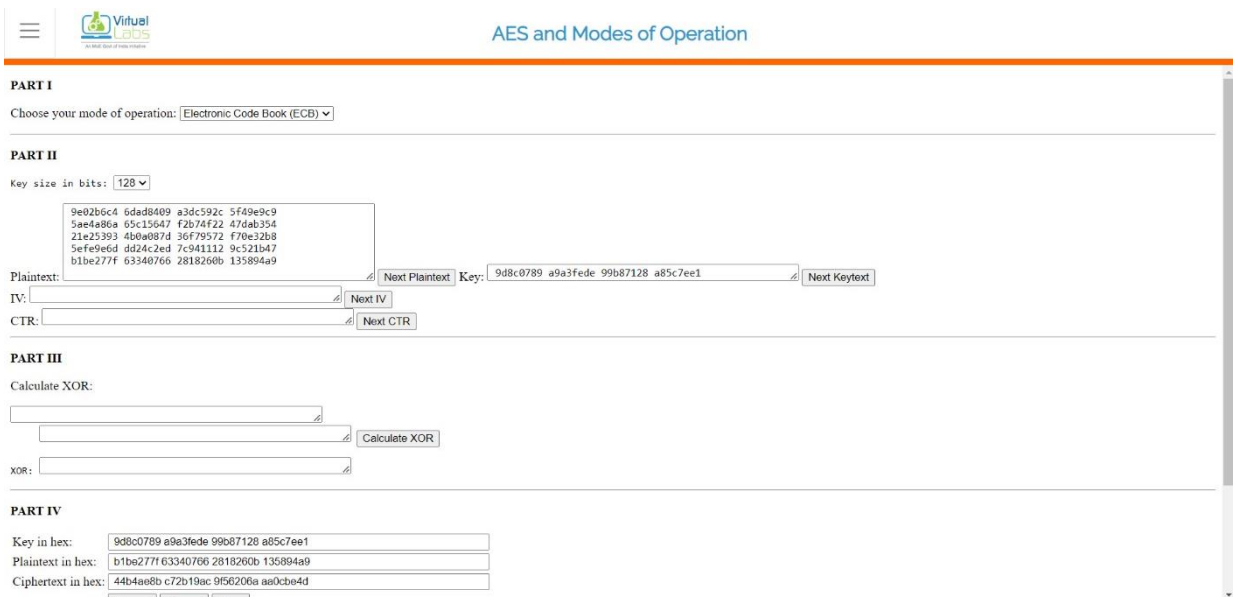
Block cipher modes of operation play a crucial role in making block ciphers practical for encrypting larger amounts of data. Each mode has its strengths and weaknesses, and the choice of mode depends on the specific requirements of the application. It's important to choose the appropriate mode based on factors such as security, performance, and desired features like error propagation or parallelizability. Always ensure you're using a well-established and properly implemented cryptographic library or tool to achieve secure data encryption.

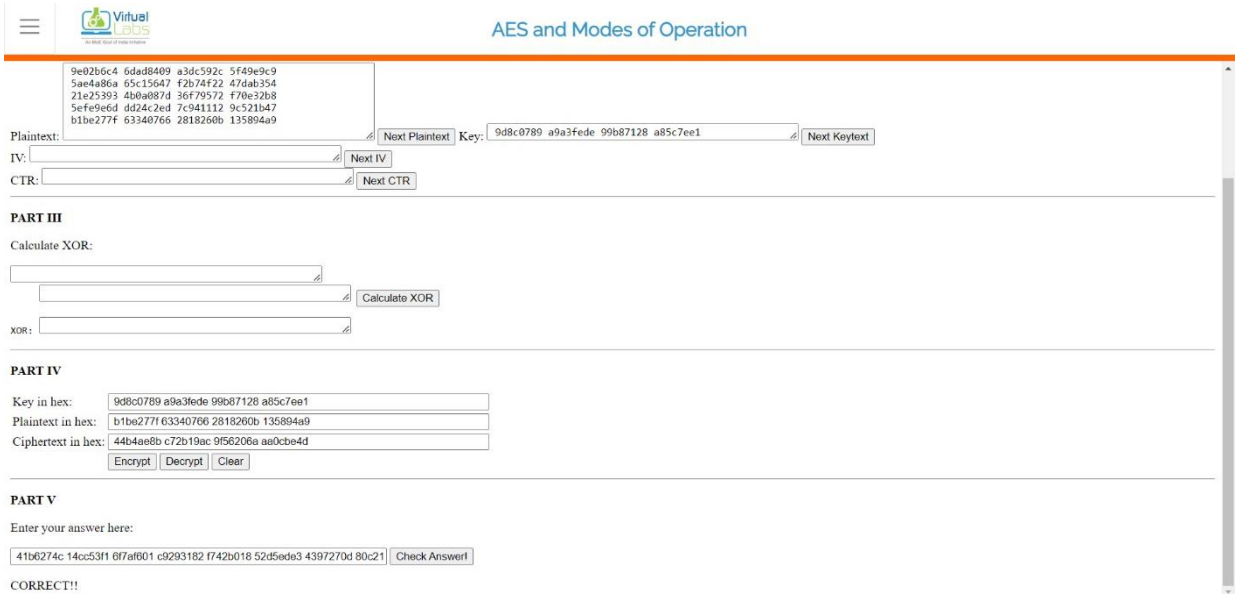
Output:

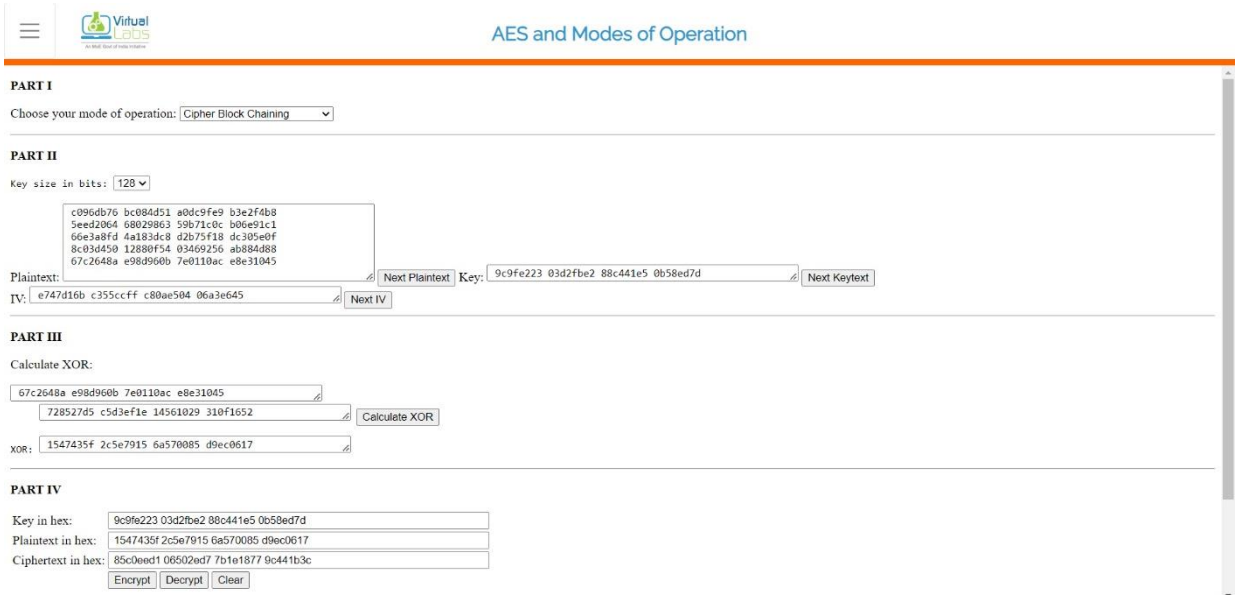


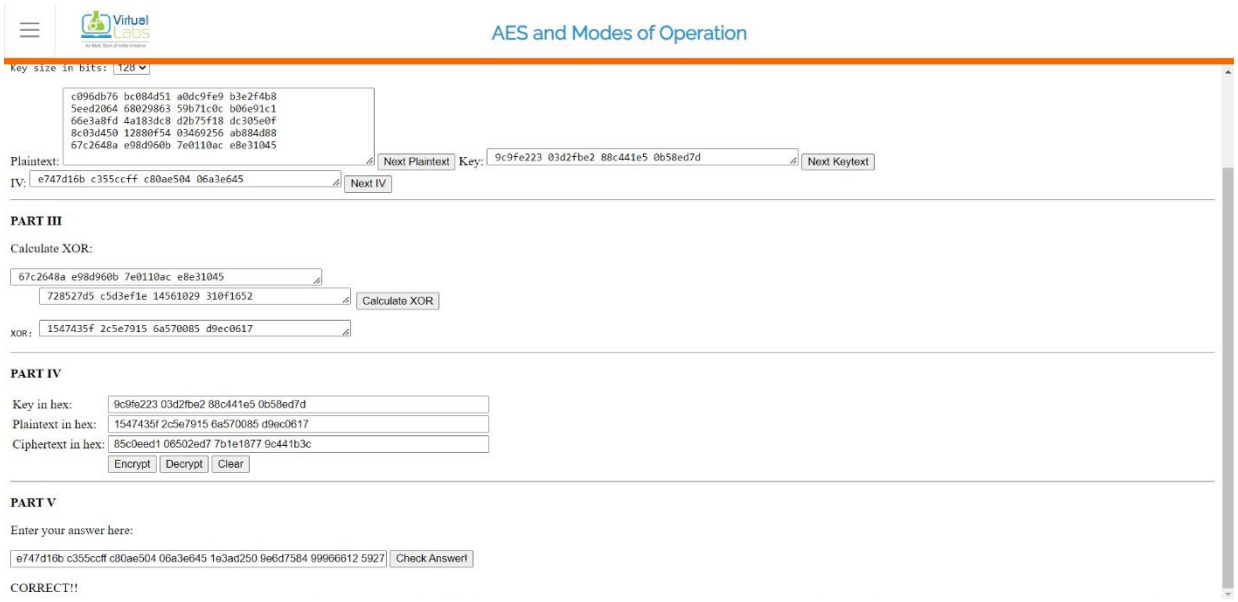


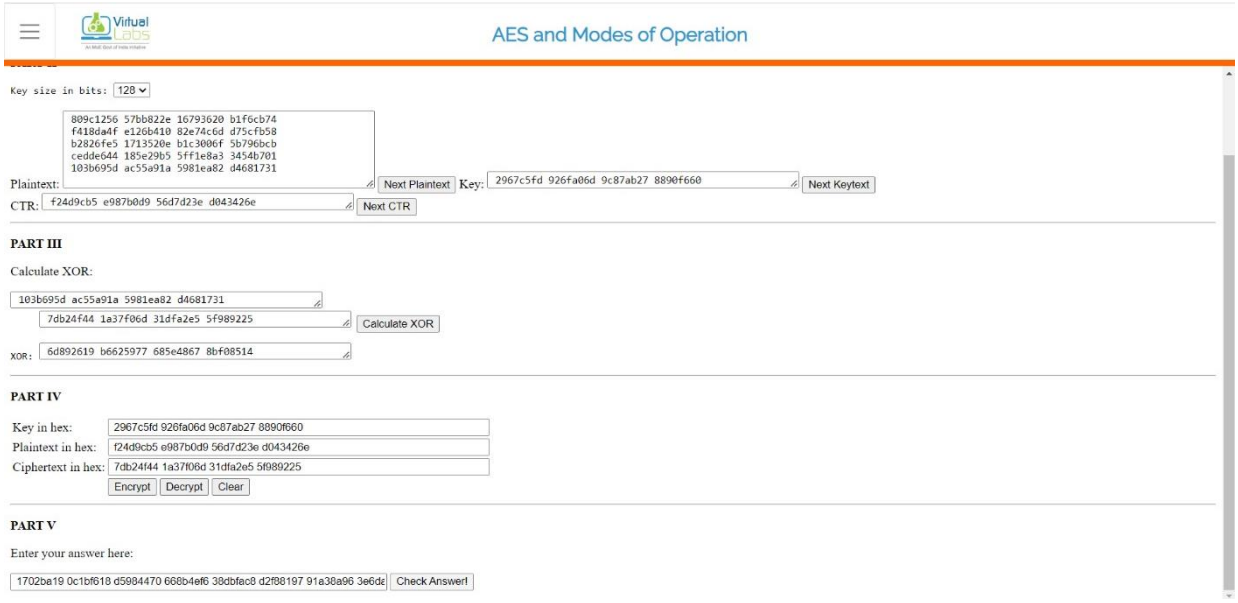

AES and Modes of Operation
Key size in bits: 128
efbfd9b5 16be4bf5 3f4a32ae 18225641
a28e6b05 f9dd6d0e 2ceb4ac6 43e0bcb0
4f4fda79 65b7567c bede510c 3feceea7
5b7befac 25904cc9 e8246988 e1c02e51
4f6a92c1 6607fca4 a1682d56 fbf0b537
Plaintext:
Next Plaintext
Key: 969827e3 18d136da cce9794a 9fe9911c
Next Keytext
IV: d7d68add bc0a6bad 4b16082b 8a62c28a
Next IV
PART III
Calculate XOR:
4f6a92c1 6607fca4 a1682d56 fbf0b537
1f6b8715 33427730 88c30c37 954c1685
Calculate XOR
XOR: 500115d4 55458b94 29ab2161 6ebca3b2
PART IV
Key in hex: 969827e3 18d136da cce9794a 9fe9911c
Plaintext in hex: 0183e3a3 5b614f98 eac112d1 16daea81
Ciphertext in hex: 1f6b8715 33427730 88c30c37 954c1685
Encrypt
Decrypt
Clear
PART V
Enter your answer here:
d7d68add bc0a6bad 4b16082b 8a62c28a 74fb98c8 11c9008d c0dedba0 a29
Check Answer!
CORRECT!!


AES and Modes of Operation
PART I
Choose your mode of operation: Electronic Code Book (ECB)
PART II
Key size in bits: 128
9e02b6c4 6dad8409 a3dc592c 5f49e9c9
5ae4a86a 65c15647 f2b74f22 47dab354
21e25393 4b0a087d 36f79572 f70e32b8
5efe9e6d dd24c2ed 7c941112 9c521b47
b1be277f 63340766 2818260b 135894a9
Plaintext:
Next Plaintext
Key: 9d8c0789 a9a3fede 99b87128 a85c7ee1
Next Keytext
IV:
Next IV
CTR:
Next CTR
PART III
Calculate XOR:
Calculate XOR
XOR:
PART IV
Key in hex: 9d8c0789 a9a3fede 99b87128 a85c7ee1
Plaintext in hex: b1be277f 63340766 2818260b 135894a9
Ciphertext in hex: 44b4ae8b c72b19ac 9f56206a aa0cbe4d


AES and Modes of Operation
9e02b6c4 6dad8409 a3dc592c 5f49e9c9
5ae4a86a 65c15647 f2b74f22 47dab354
21e25393 4b0a087d 36f79572 f70e32b8
5efe9e6d dd24c2ed 7c941112 9c521b47
b1be277f 63340766 2818260b 135894a9
Plaintext:
Next Plaintext
Key: 9d8c0789 a9a3fede 99b87128 a85c7ee1
Next Keytext
IV:
Next IV
CTR:
Next CTR
PART III
Calculate XOR:
Calculate XOR
XOR:
PART IV
Key in hex: 9d8c0789 a9a3fede 99b87128 a85c7ee1
Plaintext in hex: b1be277f 63340766 2818260b 135894a9
Ciphertext in hex: 44b4ae8b c72b19ac 9f56206a aa0cbe4d
Encrypt
Decrypt
Clear
PART V
Enter your answer here:
41b6274c 14cc53f1 6f7af601 c9293182 f742b018 52d5ede3 4397270d 80c21
Check Answer!
CORRECT!!


AES and Modes of Operation
PART I
Choose your mode of operation: Cipher Block Chaining
PART II
Key size in bits: 128
c096db76 bc084d51 a0dc9fe9 b3e2f4b8
5eed2064 68029863 59b71c0c b06e91c1
66e3a8fd 4a183dc8 d2b75f18 dc305e0f
8c03d450 12880f54 03469256 ab884d88
67c2648a e98d960b 7e0110ac e8e31045
Plaintext:
Next Plaintext
Key: 9c9fe223 03d2fbe2 88c441e5 0b58ed7d
Next Keytext
IV: e747d16b c355ccff c80ae504 06a3e645
Next IV
PART III
Calculate XOR:
67c2648a e98d960b 7e0110ac e8e31045
728527d5 c5d3ef1e 14561029 310f1652
Calculate XOR
XOR: 1547435f 2c5e7915 6a570085 d9ec0617
PART IV
Key in hex: 9c9fe223 03d2fbe2 88c441e5 0b58ed7d
Plaintext in hex: 1547435f 2c5e7915 6a570085 d9ec0617
Ciphertext in hex: 85c0eed1 06502ed7 7b1e1877 9c441b3c
Encrypt
Decrypt
Clear





Conclusion:

In conclusion, understanding block cipher modes of operation is essential for secure data encryption. Each mode offers distinct security properties and features. Careful consideration of application requirements is vital to select the most suitable mode, balancing security, performance, and desired functionalities for effective encryption practices.