Assignment 1

Aim: Shift cipher and mono alphabet substitution cipher.

Theory:

Monoalphabetic Cipher is a part of the substitution technique in which a single cipher alphabet is used per message (mapping is done from plain alphabet to cipher alphabet). Monoalphabetic cipher converts plain text into cipher text and re-convert a cipher text to plain text.

Monoalphabetic Cipher eliminates the brute-force techniques for cryptanalysis. Moreover, the cipher line can be a permutation of the 26 alphabetic characters.

Implementation:

Breaking the Mono-alphabetic Substitution Cipher

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

```
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
qwndesrhn x cevvur uxcruurq 'qhwto fr', vkr detvrtvn el gkwdk dxinr
krh ve nkhwto vee nfxuu ve hrxdk vkr orb. x dxor gwvk 'rxv fr' et wv
dxinrn krh ve pheg ve nidk x vhrfrtqein nwmr krh krxq kwvn vkr
drwuwtp.
```

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

PART II

Note that the *cipher text is in lower case* and when you replace any character, the final character of replacement, i.e., *plaintext is changed to upper case* automatically in the following scratchpad.

Scratchpad:

```
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 TO
FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN DISCOVERS
A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK
TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO
GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING.
```

Modify the text above (in scratchpad):

This is case *insensitive* function and replaces only cipher text (lower case) by plain text (upper case).

Replace cipher character m by plaintext character z 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 n by character ns 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 c by B You replaced w by I You replaced g by W You replaced e by O You replaced u by L You replaced q by D You replaced t by N You replaced ns by N You replaced n by ns You replaced n by S 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 l by F You replaced i by U You replaced b by Y You replaced f by M You replaced m by Z

PART III

Enter your solution plaintext here:

```
TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN
DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE
HER TO SHRINK TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT
CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE
CEILING.
```

Solution Key = xcdqrlpkwzoufteyahnvisgjbm

Check Answer!

CORRECT!!

Name: Pratham Asnani
Batch: T13
Roll No.-63

Assignment 3

Aim: Playfair and Vigenere cipher.

Theory:
1. Playfair 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:
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.
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’
For example:
Diagraph: "me"
Encrypted Text: cl
Encryption:
m -> c
e -> l

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair cipher -

Encode

Search for a tool
SEARCH A TOOL ON DCODE
e.g. type 'random'
BROWSE THE FULL DCODE TOOLS' LIST
Results
PRATHAM
RATHPTCV

Decode

SEARCH A TOOL ON DCODE
e.g. type 'random'
BROWSE THE FULL DCODE TOOLS' LIST
Results
PRATHAMX
lights with us
PlayFair Cipher - dCode
Tag(s) : Polygrammic Cipher, Grid Cipher
Share
dCode and more
dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!
visitors
PLAYFAIR DECODER
PLAYFAIR CIPHERTEXT
RATHPTCV
PLAYFAIR GRID
1 P R A T H
2 M B C D E
3 F G I K L
4 N O Q S U
5 V W X Y Z
RESIZE
CLEAR
PRATHMBCDEFGIKLNOQSUVWXYZ
SHIFT IF SAME ROW
Cell on the left ← (Encryption with right cell →)
SHIFT IF SAME COLUMN
Cell above ↑ (Encryption with below cell ↓)
ORDER OF LETTER ELSEWHERE
Same row as letter 1 first
DECRYPT PLAYFAIR
WITHOUT KNOWING KEY/GRID
PLAINTEXT LANGUAGE (EXPECTED)
English
DICTIONARY ATTACK (COMMON KEYS/GRIDS)
HILL CLIMBING + SIMULATED ANNEALING ATTACK (LIMITED)
KNOWN/PROBABLE PLAINTEXT ATTACK

Name-Pratham Asnani
RollNo.-63
Batch-T13

Assignment 3

Aim:-To study the concept, working, types, and various modes of operation of Block Ciphers, along with examples, to understand their application and importance in modern cryptography.

Theory:-

1. Block Cipher

1.1 Definition

A **Block Cipher** is a symmetric encryption technique that divides plaintext into fixed-size blocks (e.g., 64 or 128 bits) and encrypts each block independently using a symmetric key and an algorithm. The output is a ciphertext block of the same size.

1.2 Working Principle
Plaintext is split into equal-sized blocks.
Each block is transformed into ciphertext through multiple rounds of substitution and permutation.
The encryption and decryption processes use the same secret key.

1.3 Block Size and Key Size
Common block sizes: 64 bits (DES), 128 bits (AES)
Key sizes vary by algorithm: DES uses 56 bits, AES uses 128/192/256 bits.

1.4 Examples of Block Ciphers

Algorithm	Block Size	Key Size	Notes
DES	64 bits	56 bits	Older standard, now insecure
AES	128 bits	128 / 192 / 256 bits	Widely used, secure
Blowfish	64 bits	32 – 448 bits	Fast and flexible
IDEA	64 bits	128 bits	Resistant to cryptanalysis

1.5 Types of Block Ciphers

Feistel Network: Splits block into halves, processes halves alternately (e.g., DES)
Substitution-Permutation Network (SPN): Applies layers of substitutions and permutations (e.g., AES)

2. Modes of Operation for Block Ciphers

Block ciphers encrypt fixed-size blocks. To encrypt messages longer than one block, different **modes of operation** are used. Each mode provides specific features like error propagation, parallelism, and security properties.

2.1 Electronic Codebook (ECB) Mode

Description:

ECB mode is the simplest mode where each block of plaintext is encrypted independently using the block cipher algorithm and the same key.

Characteristics:

Each plaintext block maps directly to a ciphertext block.
Identical plaintext blocks encrypt to identical ciphertext blocks.

Advantages:

Simple and easy to implement.
Parallel encryption and decryption of blocks.

Disadvantages:

Patterns in plaintext are visible in ciphertext, leading to poor security.
Vulnerable to block replay and substitution attacks.

Use Case: Suitable only for encrypting small amounts of random data.

Electronic Codebook (ECB) mode encryption

2.2 Cipher Block Chaining (CBC) Mode
Description:

In CBC mode, each plaintext block is XORed with the previous ciphertext block before encryption. An Initialization Vector (IV) is used for the first block to ensure uniqueness.

Encryption Formula:

$C_i = E_K(P_i \oplus C_{i-1})$

where $C_0 = IV$.

Decryption Formula:

$P_i = D_K(C_i) \oplus C_{i-1}$

Characteristics:

Introduces dependency between ciphertext blocks.
Same plaintext blocks produce different ciphertext blocks due to chaining.

Advantages:

Hides patterns in plaintext effectively.
Resistant to replay attacks.

Disadvantages:

Encryption cannot be parallelized (must wait for previous block).
A single bit error in ciphertext affects the decryption of the current and next block (error propagation).

Use Case: Commonly used in secure file and data encryption.

2.3 Cipher Feedback (CFB) Mode
Description:

CFB mode turns a block cipher into a self-synchronizing stream cipher by using previous ciphertext blocks to generate the keystream.

Encryption Process:

*Ci=Pi⊕EK(Ci−1)C_i = P_i \oplus E_K(C_{i-1})*Ci =Pi ⊕EK (Ci−1)

Decryption Process:

*Pi=Ci⊕EK(Ci−1)P_i = C_i \oplus E_K(C_{i-1})*Pi =Ci ⊕EK (Ci−1)

Initialization Vector (IV) is used for *C0C_0*C0 .

Characteristics:

Can encrypt data in units smaller than the block size (e.g., bits or bytes).
Self-synchronizing — if synchronization is lost, it recovers after processing a few blocks.

Advantages:

Suitable for encrypting streams of variable length.
Error propagation limited to the block size.

Disadvantages:

Encryption is serial (cannot be parallelized).
Errors affect decryption of current and next few bits.

Use Case: Useful in communication systems and data streams.

Cipher Feedback (CFB) mode encryption

2.4 Output Feedback (OFB) Mode
Description:

OFB mode converts a block cipher into a synchronous stream cipher by generating keystream blocks independently of plaintext and ciphertext.

Keystream Generation:

$O_i = E_K(O_{i-1})$

where $O_0 = IV$.

Encryption/Decryption:

$C_i = P_i \oplus O_i$ $P_i = C_i \oplus O_i$

Characteristics:
Keystream is independent of plaintext and ciphertext.
Errors in transmission do not propagate.

Advantages:
No error propagation beyond the erroneous bit.
Encryption and decryption can be done in parallel.
Allows precomputation of keystream.

Disadvantages:

Requires unique IV for each message.
Vulnerable if IV reused.

Use Case: Ideal for noisy communication channels and real-time systems.

Output Feedback (OFB) mode encryption

2.5 Counter (CTR) Mode
Description:

CTR mode encrypts a counter value for each block and XORs it with the plaintext to produce ciphertext.

Encryption/Decryption Formula:

*Ci=Pi⊕EK(Nonce||Counteri)C_i = P_i \oplus E_K(Nonce || Counter_i)*Ci =Pi ⊕EK (Nonce||Counteri) *Pi=Ci⊕EK(Nonce||Counteri)P_i = C_i \oplus E_K(Nonce || Counter_i)*Pi =Ci ⊕EK (Nonce||Counteri)

The counter increments for each block.
Nonce is a unique number for each encryption session.

Characteristics:

Allows parallel encryption and decryption.
No error propagation.
Requires unique nonce for security.

Advantages:

Highly efficient, supports random access to encrypted data.
Easy to implement in hardware/software.

Disadvantages:

If nonce/counter repeats, security is compromised.

Use Case: High-speed encryption for disk encryption, network protocols.

Counter (CTR) mode encryption

Counter (CTR) mode encryption

3. Summary Table of Modes of Operation

Mode	Plaintext → Ciphertext Dependency	Parallelizable	Error Propagation	Use Case
ECB	None (each block independent)	Yes	No	Small data, testing
CBC	Depends on previous ciphertext	No	Yes (2 blocks)	File encryption
CFB	Depends on previous ciphertext	No	Limited	Stream data

OFB	Keystream independent	Yes	No	Noisy channels
CTR	Keystream from counter	Yes	No	High-speed encryption

4. Conclusion

Block ciphers form the backbone of symmetric key cryptography, securing data by encrypting it in fixed-size blocks. The choice of mode of operation greatly impacts the security and performance of encryption. Modes like CBC and CTR are widely adopted in practice due to their balance of security and efficiency.

Assignment- 5

Aim: Implementation and analysis of RSA cryptosystem and Digital Signature scheme using RSA.
LO-2

Theory-
Implementation and Analysis of RSA Cryptosystem and Digital Signature Scheme using RSA

The **RSA cryptosystem** is one of the most widely used public-key encryption algorithms designed for secure data transmission. It is based on the mathematical difficulty of factoring large prime numbers. The RSA algorithm involves three main steps — key generation, encryption, and decryption. In key generation, two large prime numbers are chosen to generate the public and private keys. The public key is used for encryption, while the private key is used for decryption. The strength of RSA lies in the computational infeasibility of deriving the private key from the public key, ensuring confidentiality and security.

The **Digital Signature Scheme using RSA** ensures data integrity and authenticity. Instead of encrypting the message itself, a hash of the message is created and then encrypted using the sender's private key to generate a digital signature. The receiver can verify this signature by decrypting it with the sender's public key and comparing it with the hash of the received message. If both match, it confirms that the message is authentic and has not been tampered with.

Through implementation and analysis, RSA demonstrates robust performance in securing communication, providing both **confidentiality** (via encryption) and **authentication** (via digital signatures). However, it also requires significant computational resources, especially with large key sizes, making optimization essential for practical applications.

Output-

Conclusion-

In conclusion, the implementation and analysis of the RSA cryptosystem and its digital signature scheme highlight its effectiveness in ensuring secure communication and data authenticity. Despite its computational complexity, RSA remains a cornerstone of modern cryptography due to its strong encryption, reliability, and ability to provide both confidentiality and integrity in digital transactions.

ASSIGNMENT 6 : hashing files and auditing using hashdeep

AIM: To explore hashdeep tool in kali linux for generating, matching and auditing hash of files.

LO2
THEORY:
Hashing serves the crucial purpose of ensuring data integrity, security, and efficient data retrieval.
It'sused in various applications like password storage, digital signatures, data verification, and more. Hashing generates a fixed-size output (hash value) from an input (data), making it efficient for comparing large datasets and detecting changes.

Different Hashing Algorithms:
1. MD5 (Message Digest Algorithm 5)
2. SHA-1 (Secure Hash Algorithm 1)
3. SHA-256 (Secure Hash Algorithm 256)
4. SHA-512 (Secure Hash Algorithm 512)
5. SHA-3 (Secure Hash Algorithm 3)
6. Whirlpool

Hashdeep is a command-line tool in Kali Linux used for computing and verifying file hash values, such as MD5, SHA-1, SHA-256, etc. It calculates hashes for files and directories and can create hash databases for later comparison. Hashdeep supports recursive hashing, making it useful for validating file integrity over time. It's commonly used for digital forensics, data verification, and ensuring file authenticity in security assessments.

1. Check Hashdeep Version: `hashdeep-V`
2. Display Help: `hashdeep-h` or `hashdeep-hh`
3. Manual Page: `man hashdeep`
4. Manual Page for Specific Algorithm: `man md5deep`
5. Hash a File: `hashdeep filename`
6. Hash with Hidden Paths: `hashdeep-b filename`
7. Suppress Errors: `hashdeep-s filename`
8. Multiple Hash Algorithms: `hashdeep-c md5,sha1,sha256,tiger filename`
9. Hash Multiple Files (MD5): `hashdeep-c md5 *.txt`
10. Hash Multiple Files (MD5 & SHA-1): `hashdeep-c md5,sha1 *.txt`
11. Hashing Block of Files: `hashdeep-c md5-p 100 example.txt`
12. Recursive Hashing: `hashdeep-c md5-r /home/user/myfiles`
13. Redirect Output: `md5deep *.txt > hashset.txt`
14. Matching Mode Output: `md5deep-m hashset.txt *`
15. Suppress System Messages: `md5deep-s-m hashset.txt *`
16. Display Negatively Matching Files: `md5deep-s-x hashset.txt *`

Forensic auditing can be done using hashdeep tool which means a check to determine if any files in the system are changed due to malware or any normal

system operation like update patching.
17. Create Hashset and Audit:- Create Hashset: `hashdeep-c md5,sha1,sha256-r /home/user/myfiles > hashset1.txt`- Audit Files: `hashdeep-a-r-k hashset1.txt /home/user/myfiles`
18. Audit with New File (Fails):- Create New File: `touch /home/user/myfiles/newfile.txt`- Audit Again: `hashdeep-a-r-k hashset1.txt /home/user/myfiles`
19. Check Failed Points (Verbose):- Audit with Verbose: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`
20. Audit After Moving File:- Move File: `mv /home/user/myfiles/example.txt /tmp`- Audit Again: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`
21. Audit After Renaming File:- Rename File: `mv /home/user/myfiles/shreya.txt /home/user/myfiles/backup.txt`- Audit Again: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`
22. Verbose Audit Output:- More Verbose: `hashdeep-vv-a-r-k hashset1.txt /home/user/myfiles`- Very Verbose: `hashdeep-vvv-a-r-k hashset1.txt /home/user/myfiles`

Note: Replace the paths and filenames with actual directory and file names as Needed.

OUTPUT:

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$  hashdeep c md5 -r /home/lab1006/T13-53-CNS
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb623846526884a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS/1
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS/1
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS/2
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS/2
##
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb623846526884a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS
##
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb623846526884a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
```

```
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CN5/1/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CN5/2/hashset1.txt
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CN5/1/file2.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb623846526884a3,/home/lab1006/T13-53-CN5/2/hashtext1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man hashdeep
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man md5
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep *.txt>hashset.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset4.txt
41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset4.txt
41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep *.txt>hashset5.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset5.txt
ed2bf26a0e00346cc78b1070d102897  /home/lab1006/hashset4.txt
d5d34c74e59d16bd6d5b3683db655c3  /home/lab1006/file2.txt
41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
e6e6f3b88d9c96104d0499b1b48d5c0  /home/lab1006/hashset1.txt
f26210280eb554b26753aeeb570d8bb  /home/lab1006/hashtext1.txt
fbf270dfffacfa7c55334ef6018efb7  /home/lab1006/hashset.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~$
## Invoked from: /home/lab1006
## $ hashdeep -c md5,sha1,sha256,tiger hashset.txt
##
58,1fbf270dfffacfa7c55334ef6018efb7,313fa712356dc5a57d734e4328976002d2bd413a,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,25d855fccd1f93f7049d0d85ac
e4ac5b827325c929850e12,/home/lab1006/hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 *.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 file2.txt hashset1.txt hashset.txt hashtext1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file2.txt
58,1fbf270dfffacfa7c55334ef6018efb7,/home/lab1006/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,/home/lab1006/hashset1.txt
370,6f26210280eb554b26753aeeb570d8bb,/home/lab1006/hashtext1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha256 *.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5,sha256 file2.txt hashset1.txt hashset.txt hashtext1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/file2.txt
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/hashset1.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb623846526884a3,/home/lab1006/hashtext1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -b -c md5 *.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -b -c md5 file2.txt hashset1.txt hashset.txt hashtext1.txt
##
0,d41d8cd98f00b204e9800998ecf8427e,file2.txt
58,1fbf270dfffacfa7c55334ef6018efb7,hashset.txt
370,6f26210280eb554b26753aeeb570d8bb,hashtext1.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

Conclusion-

Hashing with **hashdeep** produces fast, repeatable cryptographic checksums across many files, making it ideal for integrity verification and digital-forensics evidence preservation. With support for multiple algorithms and recursive comparison/lookup features, it simplifies detecting tampering, auditing large datasets, and automating file-integrity checks.

Assignment 7

Aim:-
Study the use of network reconnaissance tools like WHOSI ,dig ,traceroute ,nslookup ,nikto ,dmitry to gather information about networks and domain registars

WHOIS

Whois is a command-line utility used in Linux systems to retrieve information about domain names, IP addresses, and network devices registered with the Internet Corporation for Assigned Names and Numbers (ICANN). The data received by Whois consists of the name and contact information of the domain or IP address owner, the registration and expiration date, the domain registrar, and the server information. Whois command can be very useful for network administrators, web developers, and security professionals for achieving various tasks like checking network connectivity or troubleshooting. In this article, we will go through the usage of the Whois command on Linux (Ubuntu system).

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ whois geeksforgeeks.org
Domain Name: geeksforgeeks.org
Registry Domain ID: f507261c73964021b7430545a8b370ee-LROR
Registrar WHOIS Server: http://whois.publicdomainregistry.com
Registrar URL: http://www.publicdomainregistry.com
Updated Date: 2022-04-21T06:36:07Z
Creation Date: 2009-03-19T06:08:55Z
Registry Expiry Date: 2030-03-19T06:08:55Z
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Registrar Abuse Contact Email: abuse@publicdomainregistry.com
Registrar Abuse Contact Phone: +1.2013775952
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID: REDACTED
Registrant Name: REDACTED
Registrant Organization: Privacy Protect, LLC (PrivacyProtect.org)
Registrant Street: REDACTED
Registrant City: REDACTED
Registrant State/Province: MA
Registrant Postal Code: REDACTED
Registrant Country: US
Registrant Phone: REDACTED
Registrant Phone Ext: REDACTED
Registrant Fax: REDACTED
Registrant Fax Ext: REDACTED
Registrant Email: REDACTED
Registry Admin ID: REDACTED
Admin Name: REDACTED
Admin Organization: REDACTED
Admin Street: REDACTED
Admin City: REDACTED
Admin State/Province: REDACTED
Admin Postal Code: REDACTED
Admin Country: REDACTED
Admin Phone: REDACTED
Admin Phone Ext: REDACTED
Admin Fax: REDACTED
Admin Fax Ext: REDACTED
Admin Email: REDACTED
Registry Tech ID: REDACTED
Tech Name: REDACTED
Tech Organization: REDACTED
Tech Street: REDACTED
Tech City: REDACTED
Tech State/Province: REDACTED
```

dig

dig command stands for ***Domain Information Groper***. It retrieves information about DNS name servers. Network administrators use it to verify and troubleshoot DNS problems and perform DNS lookups. The dignslookup and the host.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dig google.com

; <<>> DiG 9.11.3-1ubuntu1.18-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18563
;; 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.             262     IN      A       142.251.43.14

;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Thu Jul 24 15:07:50 IST 2025
;; MSG SIZE  rcvd: 55
```

nslookup

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS-related problems.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.43.14
Name:   google.com
Address: 2404:6800:4009:809::200e
```

traceroute

The `**traceroute**` command is a network diagnostic tool used to trace the route taken by packets from a source to a destination over an IP network. It provides valuable insights into the network path, including the number of hops (routers) between the source and destination and the round-trip time (RTT) for each hop.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute www.goole.com
traceroute to www.goole.com (217.160.0.201), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.655 ms  0.614 ms  5.373 ms
 2  203.212.25.1 (203.212.25.1)  2.165 ms  2.275 ms  2.097 ms
 3  203.212.24.53 (203.212.24.53)  2.207 ms  2.297 ms  2.245 ms
 4  10.10.226.153 (10.10.226.153)  3.110 ms  3.620 ms  3.566 ms
 5  * * *
 6  203.200.11.141.ill-bgl.static.vsnl.net.in (203.200.11.141)  4.257 ms * *
 7  172.28.118.237 (172.28.118.237)  3.655 ms  3.585 ms *
 8  * * *
 9  if-ae-2-2.tcore2.mlv-mumbai.as6453.net (180.87.38.2)  192.463 ms if-ae-32-2.tcore2.mlv-mumbai.as6453.net (180.87.38.147)  189.349 ms  189.342 ms
10  * * *
11  * * *
12  195.219.213.137 (195.219.213.137)  129.937 ms  129.922 ms  129.983 ms
13  ldn-bb1-link.ip.twelve99.net (62.115.140.72)  132.854 ms ldn-bb2-link.ip.twelve99.net (62.115.140.70)  122.780 ms ldn-bb1-link.ip.twelve99.net (62.115.120.74)  122.
755 ms
14  prs-bb2-link.ip.twelve99.net (62.115.133.239)  128.429 ms  127.964 ms  127.780 ms
15  ffm-bb2-link.ip.twelve99.net (62.115.122.139)  141.139 ms  138.163 ms ffm-bb1-link.ip.twelve99.net (62.115.123.12)  129.086 ms
16  * * *
17  ionos-ic-350360.ip.twelve99-cust.net (62.115.181.11)  131.592 ms  129.463 ms  129.378 ms
18  lo-0.rc-b.bs.kae.de.net.ionos.com (212.227.117.207)  129.845 ms  130.239 ms  129.430 ms
19  lo-0.gw-distd-sh-1.bs.kae.de.net.ionos.com (212.227.112.252)  130.140 ms lo-0.gw-distd-sh-2.bs.kae.de.net.ionos.com (212.227.112.253)  138.687 ms  138.659 ms
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Nikto

Nikto is an open-source web server scanner which performs comprehensive tests against web servers for multiple items. You can use Nikto with any web servers like Apache, Nginx, IHS, OHS, Litespeed, and so on. Nikto can check for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Items and plugins scanned by Nikto are frequently updated and can be automatically updated

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h google.com
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          142.251.43.14
+ Target Hostname:    google.com
+ Target Port:        80
+ Start Time:         2025-07-24 15:32:33 (GMT5.5)
---------------------------------------------------------------------------
+ Server: gws
+ Uncommon header 'content-security-policy-report-only' found, with contents: object-src 'none';base-uri 'self';script-src 'nonce-3Ag21l5IuFZbmPsMF5lTPQ' 'strict-dynami
c' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
+ Uncommon header 'x-xss-protection' found, with contents: 0
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN
+ Root page / redirects to: http://www.google.com/
+ Uncommon header 'referrer-policy' found, with contents: no-referrer
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server banner has changed from 'gws' to 'sffe' which may suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'cross-origin-resource-policy' found, with contents: cross-origin
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ Cookie AEC created without the httponly flag
+ Cookie NID created without the httponly flag
+ Uncommon header 'content-security-policy' found, with contents: object-src 'none';base-uri 'self';script-src 'nonce-Vvypc2knlVm2iE_1uvkrfQ' 'strict-dynamic' 'report-s
ample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other
+ Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Model, Sec-CH-UA-W
oW64, Sec-CH-UA-Form-Factors, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version
+ Uncommon header 'permissions-policy' found, with contents: ch-ua-arch=*, ch-ua-bitness=*, ch-ua-full-version=*, ch-ua-full-version-list=*, ch-ua-model=*, ch-ua-wow64=
*, ch-ua-form-factors=*, ch-ua-platform=*, ch-ua-platform-version=*
+ Uncommon header 'cross-origin-opener-policy' found, with contents: same-origin
```

dmitry

Dmitry is a free and open-source tool available on GitHub. The tool is used for information gathering. You can download the tool and install in your Kali Linux. Dmitry stands for DeepMagic Information Gathering Tool. It's a command-line tool Using Dmitry tool You can collect information about the target, this information can be used for social engineering attacks. It can be used to gather a number of valuable pieces of information

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dmitry google.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:142.251.43.14
HostName:google.com

Gathered Inet-whois information for 142.251.43.14
---------------------------------

inetnum:        142.248.0.0 - 143.13.255.255
netname:        NON-RIPE-NCC-MANAGED-ADDRESS-BLOCK
descr:          IPv4 address block not managed by the RIPE NCC
remarks:        -------------------------------------------------------
remarks:
remarks:        For registration information,
remarks:        you can consult the following sources:
remarks:
remarks:        IANA
remarks:        http://www.iana.org/assignments/ipv4-address-space
remarks:        http://www.iana.org/assignments/iana-ipv4-special-registry
remarks:        http://www.iana.org/assignments/ipv4-recovered-address-space
remarks:
remarks:        AFRINIC (Africa)
remarks:        http://www.afrinic.net/ whois.afrinic.net
remarks:
remarks:        APNIC (Asia Pacific)
remarks:        http://www.apnic.net/ whois.apnic.net
remarks:
remarks:        ARIN (Northern America)
remarks:        http://www.arin.net/ whois.arin.net
remarks:
remarks:        LACNIC (Latin America and the Carribean)
remarks:        http://www.lacnic.net/ whois.lacnic.net
remarks:
remarks:        -------------------------------------------------------
country:        EU # Country is really world wide
admin-c:        IANA1-RIPE
tech-c:         IANA1-RIPE
status:         ALLOCATED UNSPECIFIED
mnt-by:         RIPE-NCC-HM-MNT
created:        2025-04-16T11:49:16Z
last-modified:  2025-04-16T11:49:16Z
source:         RIPE

role:           Internet Assigned Numbers Authority
```

Assignment 8

Aim: To install Nmap and use it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP port scan, UDP portscan, etc.

Theory:

Nmap:

Nmap (Network Mapper) is a widely used open-source network scanning tooldesigned to discover hosts and services on a computer network. Its primary functions include mapping network infrastructure, identifying active devices,
and detecting vulnerabilities. It operates by sending specially crafted packets totarget hosts and analyzing their responses.

Key Features

1. Port Scanning:

- nmap identifies open ports on a target machine. By probing various ports, it determines which are accessible and responsive, which can help in assessing the exposure of services.

2. Service Detection:

- Beyond detecting open ports, nmap can probe those ports to identify the services running on them. This includes determining the version of the service, which helps in identifying potential vulnerabilities specific to that version.

3. Operating System Detection:

- Using techniques like TCP/IP stack fingerprinting, nmap estimates the operating system of the target device. It analyzes responses to certain packets to deduce characteristics of the OS.

4. Network Mapping:

- nmap can map out a network by discovering which hosts are active and how they are interconnected. This can be useful for understanding network architecture and detecting unauthorized devices.

5. Vulnerability Scanning:

- Through the use of the Nmap Scripting Engine (NSE), nmap can run scripts to check for specific vulnerabilities or security issues on the targetsystems.

Common Scanning Techniques

1. TCP SYN Scan (Stealth Scan):

- Sends SYN packets to target ports. If a SYN-ACK is received, the port is open. This method is less likely to be logged by the target's firewall or intrusion detection system.

2. TCP Connect Scan:

- Completes the TCP handshake to determine open ports. This scan is more intrusive and detectable compared to the SYN scan but can be more accurate.

3. UDP Scan:

- Sends UDP packets to ports and waits for responses. Since UDP is connectionless, it can be less reliable and slower to scan compared toTCP.

4. Aggressive Scan:

- Combines several scans and techniques (OS detection, version detection, script scanning) into a single scan, providing a comprehensive overview but potentially more detectable.

Practical Applications

• Security Assessment:

- Used by security professionals to find vulnerabilities, misconfigurations, and security holes in a network.

Shrutik Gupta

• Network Inventory:

- Helps network administrators maintain an inventory of devices and their services, ensuring network hygiene.

• Troubleshooting:

- Assists in diagnosing network issues by revealing misconfigured or malfunctioning services.

Nmap Installation:

Shrutik Gupta

Microsoft Windows binaries

Please read the Windows section of the Install Guide for limitations and installation instructions for the Windows version of Nmap. It's provided as an executable self-installer which includes Nmap's dependencies and the Zenmap GUI. We support Nmap on Windows 7 and newer, as well as Windows Server 2008 R2 and newer. We also maintain a guide for users who must run Nmap on earlier Windows releases..

Latest stable release self-installer: nmap-7.95-setup.exe

We have written post-install usage instructions. Please notify us if you encounter any problems or have suggestions for the installer.

Nmap Setup
Choose Components
Choose which features of Nmap you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Nmap Core Files
Register Nmap Path
Npcap 1.79
Check online for newe
Network Performance
Zenmap (GUI Fronten
Ndiff (Scan comparisor
Ncat (Modern Netcat r
Space required: 80.5 MB
Description
Modifies Windows registry values to improve TCP connect scan performance. Recommended.
Nullsoft Install System v3.08
< Back
Next >
Cancel

Shrutik

Nmap Setup
Choose Install Location
Choose the folder in which to install Nmap.
Setup will install Nmap in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files (x86)\Nmap
Browse...
Space required: 80.5 MB
Space available: 248.9 GB
Nullsoft Install System v3.08
< Back
Install
Cancel

Npcap 1.79 Setup
NMAP.ORG
License Agreement
Please review the license terms before installing Npcap 1.79.
Press Page Down to see the rest of the agreement.
NPCAP COPYRIGHT / END USER LICENSE AGREEMENT
Npcap (https://npcap.com) is a Windows packet sniffing driver and library and is copyright (c) 2013-2023 by Nmap Software LLC ("The Nmap Project"). All rights reserved.
Even though Npcap source code is publicly available for review, it is not open source software and may not be redistributed or used in other software without special permission from the Nmap Project. The standard (free) version is usually limited to installation on five
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Npcap 1.79.
Nullsoft Install System v3.07
I Agree
Cancel

Shrutik

Nmap Setup
Installing
Please wait while Nmap is being installed.
Extract: Uninstall.exe... 90%
Show details
Nullsoft Install System v3.08
< Back
Next >
Cancel

Shrutik

Npcap 1.79 Setup
NMAP.ORG
Installation Complete
Setup was completed successfully.
Completed
Show details
Nullsoft Install System v3.07
< Back
Next >
Cancel

Shrutik

Commands:

Shrutik

1. Scan Open Ports

```
C:\Program Files (x86)\Nmap>nmap 162.241.70.62
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-19 09:56 India Standard Time
Nmap scan report for 162-241-70-62.webhostbox.net (162.241.70.62)
Host is up (0.26s latency).
Not shown: 915 filtered tcp ports (no-response), 15 filtered tcp ports (host-prohibited), 59 closed tcp ports (res
et)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 10.32 seconds
```

2. OS Fingerprinting

```
C:\Program Files (x86)\Nmap>nmap -O 162.241.70.62
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-19 09:56 India Standard Time
Nmap scan report for 162-241-70-62.webhostbox.net (162.241.70.62)
Host is up (0.26s latency).
Not shown: 915 filtered tcp ports (no-response), 15 filtered tcp ports (host-prohibited), 59 closed tcp ports (res
et)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql
Aggressive OS guesses: Linux 4.19 - 5.15 (92%), Linux 3.4 (91%), Linux 4.15 (90%), Linux 3.11 - 4.9 (88%), Linux 3
.2 - 3.8 (88%), Android TV OS 11 (Linux 4.19) (86%), Linux 2.6.32 (86%), Linux 2.6.32 or 3.10 (86%), Linux 2.6.39
(86%), Linux 3.10 - 3.12 (86%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.87 seconds
```

3. Ping Scan

4. TCP Port Scan

```
C:\Program Files (x86)\Nmap>nmap -sT 162.241.70.62
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-19 10:05 India Standard Time
Nmap scan report for 162-241-70-62.webhostbox.net (162.241.70.62)
Host is up (0.26s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 14.82 seconds
```

5. UDP Port Scan

```
C:\Program Files (x86)\Nmap>nmap -sU 142.250.217.68
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-19 10:24 India Standard Time
Stats: 0:03:35 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.40% done; ETC: 10:29 (0:01:10 remaining)
Stats: 0:03:35 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.55% done; ETC: 10:29 (0:01:09 remaining)
Nmap scan report for sea09s29-in-f4.1e100.net (142.250.217.68)
Host is up (0.26s latency).
Not shown: 999 open|filtered udp ports (no-response)
PORT      STATE  SERVICE
33459/udp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 222.04 seconds
```

Conclusion : LO4 achieved.

Assignment 9

Aim : To simulate DOS attack using Hping3.

LO-5
Theory :
Understanding Denial of Service (DoS) Attack
A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning and availability of a network, system, service, or resource. The primary goal of a DoS attack is to overwhelm the target with a flood of traffic or to exploit vulnerabilities in such a way that it becomes inaccessible to its legitimate users.

Let's delve deeper into some common types of DoS attacks:
SYN Flood Attack

A SYN Flood Attack is a sophisticated form of DoS attack that exploits the way TCP (Transmission Control Protocol) connections are established. In a standard TCP handshake, when a client wants to establish a connection with a server, it sends a SYN (synchronize) packet to initiate the connection. The server responds with a SYN-ACK (synchronize acknowledge) packet, and the client completes the handshake with an ACK (acknowledge) packet.

In a SYN flood attack, the attacker sends an excessive number of SYN packets to the target server but does not follow up with ACK packets to complete the handshake. Instead, the attacker continually sends new SYN packets, causing the server to allocate resources for incomplete connections. Over time, these halfopen connections can accumulate and exhaust the server's resources, making it unable to respond to legitimate connection requests. This effectively denies service to legitimate users.

ICMP Flood Attack (Ping Flood Attack)

An ICMP Flood Attack, commonly known as a Ping Flood Attack, targets the Internet Control Message Protocol (ICMP). ICMP is used for various network diagnostic purposes, including the famous "ping" utility, which checks the reachability of a network host. In a Ping Flood Attack, the attacker sends an overwhelming number of ICMP echo-request packets (ping requests) to a target Device.

The target device, as perstandard ICMP behavior,respondsto each incoming echorequest with an echo-reply. In the case of a flood attack, the attacker's goal is to generate an excessive number of echo-requests, forcing the target to respond with an equal number of echo-replies. This massive traffic can quickly consume the target's network and computing resources, causing it to become unresponsive to legitimate network traffic.

SMURF Attack

A SMURF attack is a type of Denial of Service (DoS) attack that targets the Internet Control Message Protocol (ICMP) and leverages a technique called "amplification." In a SMURF attack, the attacker sends a large number of ICMP echo-request packets (commonly known as "pings") to an intermediate network, which then reflects these packets to a victim's IP address. This results in a flood of responses overwhelming the victim's network and causing a DoS condition.
Here's how a SMURF attack works:

1. The attacker sends a large number of ICMP echo-request packets (pings) to the broadcast address of an intermediate network.
2. The routers on the intermediate network, as per standard behavior, broadcast these ICMP requests to all hosts on the network.
3. Numerous hosts on the intermediate network respond to these ICMP requests by sending ICMP echo-reply packets to the source IP address specified in the requests. Since the source IP address in the requestsis the victim's IP address, these responses flood the victim's network.
4. The victim's network becomes overwhelmed with ICMP traffic, leading to high resource utilization and unavailability of services, effectively causing a DoS Condition.

Hping3 Commands for SYN Flood and ICMP Flood
SYN Flood using Hping3

```bash hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
```
- `-c 15000`: Specifies sending 15,000 packets.
- `-d 120`: Sets the data size to 120 bytes in each packet.
- `-S`: Sets the SYN flag in TCP packets.
- `-w 64`: Defines a window size of 64.
- `-p 80`: Targets port 80 (commonly used for HTTP).
- `--flood`: Floods the target with packets continuously.
- `--rand-source`: Utilizes random source IP addresses.
- `192.168.1.159`: Specifies the target IP address.

ICMP Flood using Hping3
```bash hping3 -1 --flood -a 192.168.103 192.168.1.255
```
- `-1`: Indicates the use of ICMP echo (ping) requests.
- `--flood`: Initiates the continuous flooding of the target.
- `-a 192.168.103`: Spoofs the source IP address as 192.168.103.
- `192.168.1.255`: Targets the broadcast address, causing multiple devices on the network to respond.

Example Hping3 Command for a SMURF Attack

In a SMURF attack, Hping3 can be used to generate ICMP echo-request packets and send them to the broadcast address of an intermediate network, causing

amplification and flooding. Below is an example command:
```bash hping3 -1 --flood -a <spoofed_source_ip> <broadcast_address>
```
- `-1`: Indicates the use of ICMP echo (ping) requests.
- `--flood`: Initiates the continuous flooding of the target.
- `-a <spoofed_source_ip>`: Spoofs the source IP address as `<spoofed_source_ip>`. The attacker often uses a spoofed IP address to hide their identity.
- `<broadcast_address>`: Specifies the broadcast address of the intermediate network. This is where the ICMP requests are sent

```
altaf@LAPTOP-DGNIK4U9: ~
Preparing to unpack .../hping3_3.a2.ds2-10_amd64.deb ...
Unpacking hping3 (3.a2.ds2-10) ...
Setting up hping3 (3.a2.ds2-10) ...
Processing triggers for man-db (2.10.2-1) ...
altaf@LAPTOP-DGNIK4U9:~$ man hping3
altaf@LAPTOP-DGNIK4U9:~$ man hping3
altaf@LAPTOP-DGNIK4U9:~$ hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[sudo] password for altaf:
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
352510 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -c 15000 -d 120 -S w 64 -p 80 --flood --rand-source 192.168.1.159
[sudo] password for altaf:
hping3: you must specify only one target host at a time
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -c 15000 -d 120 w 64 -p 80 --flood --rand-source 192.168.1.159
hping3: you must specify only one target host at a time
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -c 15000 -d 120 -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): NO FLAGS are set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.159 hping statistic ---
48761 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.159 hping statistic ---
232030 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
altaf@LAPTOP-DGNIK4U9:~$ ^C
altaf@LAPTOP-DGNIK4U9:~$ ^C
altaf@LAPTOP-DGNIK4U9:~$ ^C
altaf@LAPTOP-DGNIK4U9:~$ ^C
```

```
altaf@LAPTOP-DGNIK4U9: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  hping3
0 upgraded, 1 newly installed, 0 to remove and 140 not upgraded.
Need to get 106 kB of archives.
After this operation, 263 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 hping3 amd64 3.a2.ds2-10 [106 kB]
Fetched 106 kB in 2s (64.4 kB/s)
Selecting previously unselected package hping3.
(Reading database ... 53952 files and directories currently installed.)
Preparing to unpack .../hping3_3.a2.ds2-10_amd64.deb ...
Unpacking hping3 (3.a2.ds2-10) ...
Setting up hping3 (3.a2.ds2-10) ...
Processing triggers for man-db (2.10.2-1) ...
altaf@LAPTOP-DGNIK4U9:~$ man hping3
altaf@LAPTOP-DGNIK4U9:~$ man hping3
altaf@LAPTOP-DGNIK4U9:~$ hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket
altaf@LAPTOP-DGNIK4U9:~$ sudo hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[sudo] password for altaf:
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
352510 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
altaf@LAPTOP-DGNIK4U9:~$
```

Conclusion-

A Denial-of-Service (DoS) attack disrupts targeted services by overwhelming network resources or applications, causing downtime and loss of availability for legitimate users. Preventing and mitigating DoS requires layered defenses—rate limiting, traffic filtering, and scalable infrastructure—alongside continuous monitoring and incident response planning.

Assignment 11 : IDS USING SNORT

Aim : To install snort, configuring it in Intrusion Detection mode and writing rules for detecting pinging activity.

LO-6
Theory :

1. Installing Snort
● Installation: Snort is available for both Linux and Windows. The installation involves downloading the Snort package from its official source and following the setup process. During installation, you specify the network interface that Snort will monitor.
2. Adding Rules
● Rules: Snort uses predefined rules to detect specific types of network activity that could indicate malicious behavior. These rules define patterns, actions to take (such as logging or alerting), and the traffic to inspect. Users can create custom rules or use communitycontributed rule sets.
● Structure: A Snort rule consists of an action (alert, log, etc.), protocol, source/destination IP addresses, ports, and specific options that define the detection logic.
3. Configuring Snort
● Configuration File: The main Snort configuration file specifies the network variables, rule paths, and preprocessors (used for advanced traffic detection). It also defines how Snort handles and logs alerts and what traffic patterns to monitor (such as internal vs. external networks).
● Preprocessors: These are modular add-ons that extend Snort's capabilities, enabling it to detect various network anomalies, such as port scanning or fragmented packets.
4. Validating Configuration
● Validation: Before running Snort, it is important to validate the configuration to ensure
that there are no syntax errors or misconfigurations. This process checks the integrity of the configuration file and ensures all rules and preprocessors are correctly set up.
5. Monitoring for Intrusions
● Running Snort in IDS Mode: Once Snort is configured, it can be run in intrusion detection mode. In this mode, Snort monitors network traffic in realtime and checks for matches against the active rule sets. When malicious
traffic is detected, Snort generates alerts.
● Alerting and Logging: Snort can be configured to log alerts in various formats, such as text files or centralized logging systems. Alerts can be displayed on the console or sent to external logging services for further analysis.
6. Monitoring and Analyzing Logs
● Log Review: Regular log monitoring is crucial for intrusion detection. Administrators can analyze logs manually or use web-based interfaces to visualize and manage alerts more effectively.

● Integration with Tools: For more efficient monitoring, Snort can be integrated with visualization and reporting tools like Snorby or BASE, which provide a graphical interface for analyzing intrusion alerts and trends over time.
This process provides a robust way to detect and respond to network-based attacks using Snort IDS.

Output:
sudo gedit /etc/snort/snort.conf

```
Open ▾                    *snort.conf                    Save
                          /etc/snort
42 #          for a specific interface
43 ##################################################
44 #
45 # If you want to run Snort in Debian using different
46 # instances each handling a different interface and
47 # a different configuration you can copy this file to
48 # /etc/snort/snort.$interface.conf (where '$interface' is the name of your
49 # network interface) and adjust the valuse there.
50 #
51 # The Debian init.d script is defined in such a way
52 # that you can run multiple instances.
53
54 ##################################################
55 # Step #1: Set the network variables.  For more information, see README.variables
56 ##################################################
57
58 # Setup the network addresses you are protecting
59 #
60 # Note to Debian users: this value is overriden when starting
61 # up the Snort daemon through the init.d script by the
62 # value of DEBIAN_SNORT_HOME_NET s defined in the
63 # /etc/snort/snort.debian.conf configuration file
64 #
65 ipvar HOME_NET 192.168.44.0/24
66
67 # Set up the external network addresses. Leave as "any" in most situations
68 ipvar EXTERNAL_NET any
69 # If HOME_NET is defined as something other than "any", alternative, you can
70 # use this definition if you do not want to detect attacks from your internal
71 # IP addresses:
72 #ipvar EXTERNAL_NET !$HOME_NET
73
74 # List of DNS servers on your network
75 ipvar DNS_SERVERS $HOME_NET
76
77 # List of SMTP servers on your network
78 ipvar SMTP_SERVERS $HOME_NET
79
80 # List of web servers on your network
81 ipvar HTTP_SERVERS $HOME_NET
82
83 # List of sql servers on your network
84 ipvar SQL_SERVERS $HOME_NET
```

```
shawn@Shawn-Laptop: ~
shawn@Shawn-Laptop: $ sudo snort -T -c /etc/snort/snort.conf -i eth0
[sudo] password for shawn:
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001
7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091
9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 69
88 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 90
80 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort/snort_dynamicpreprocessor/...
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dce2_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_reputation_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ftptelnet_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_gtp_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ssl_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_modbus_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_sip_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ssh_preproc.so... done
```

```
shawn@Shawn-Laptop: ~
shawn@Shawn-Laptop:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/shawn/.gnupg' created
gpg: keybox '/home/shawn/.gnupg/pubring.kbx' created
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: Shawn
Email address: dcostashawn2004@gmail.com
You selected this USER-ID:
    "Shawn <dcostashawn2004@gmail.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: /home/shawn/.gnupg/trustdb.gpg: trustdb created
gpg: key 5B010A70B6EDA437 marked as ultimately trusted
gpg: directory '/home/shawn/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/shawn/.gnupg/openpgp-revocs.d/AD71B169E4D32C9B12F457905B010A70B6EDA437.rev'
public and secret key created and signed.

pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      AD71B169E4D32C9B12F457905B010A70B6EDA437
uid                      Shawn <dcostashawn2004@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]

shawn@Shawn-Laptop:~$
```

```
+- [ Rules ] -------------------------------------------------------------
Rule application order: pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
WARNING: flowbits key 'smb.tree.create.llsrpc' is set but not ever checked.
WARNING: flowbits key 'ms_sql_seen_dns' is checked but not ever set.
33 out of 1024 flowbits in use.

[ Port Based Pattern Matching Memory ]
+- [ Aho-Corasick Summary ] ------------------------------------
| Storage Format    : Full-Q
| Finite Automaton  : DFA
| Alphabet Size     : 256 Chars
| Sizeof State      : Variable (1,2,4 bytes)
| Instances         : 215
|     1 byte states : 204
|     2 byte states : 11
|     4 byte states : 0
| Characters        : 64755
| States            : 31951
| Transitions       : 863868
| State Density     : 10.6%
| Patterns          : 5041
| Match States      : 3836
| Memory (MB)       : 16.90
|   Patterns        : 0.51
|   Match Lists     : 1.01
|   DFA
|     1 byte states : 1.02
|     2 byte states : 13.96
|     4 byte states : 0.00
+-------------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 1038 ]
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o" )~    Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>

Snort successfully validated the configuration!
Snort exiting
shawn@Shawn-Laptop: $ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
^C
^[[28~*** Caught Int-Signal
shawn@Shawn-Laptop: $ nmap 192.168.44.128
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-12 18:00 UTC
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.05 seconds
```

Conclusion-

In conclusion, Snort serves as a powerful and flexible Intrusion Detection System (IDS) that effectively monitors network traffic for suspicious activity. Its open-source nature and strong rule-based detection make it a reliable tool for enhancing network security.

Name-Pratham Asnani
RollNo.-63
Batch-T13

Assignment-10

Aim-To study and configure Firewall using IP Tables
Theory-

A **firewall** is a security system—either hardware, software, or a combination of both—that monitors and controls network traffic. Its main purpose is to create a barrier between a **trusted internal network** (like your computer or organization's LAN) and an **untrusted external network** (like the internet). By doing this, it helps prevent unauthorized access, cyberattacks, and data breaches while allowing safe communication.

`iptables` is a command-line utility in Linux used to configure the system's firewall rules. It filters and controls incoming, outgoing, and forwarded network traffic based on IP addresses, ports, and protocols. It works by applying rules to packets at different stages of their journey through the system, ensuring network security and traffic management. Administrators use iptables to prevent unauthorized access, allow trusted communication, and log network activities. It is built on the Netfilter framework of the Linux kernel.

Types of Firewall

1. **Packet-Filtering Firewall** – Examines packets individually, allowing or blocking them based on IP, port, or protocol.

2. **Stateful Inspection Firewall** – Tracks the state of active connections and makes decisions based on traffic context, not just individual packets.

3. **Application-Level Firewall (Proxy Firewall)** – Works at the application layer, filtering traffic for specific applications like HTTP or FTP.

4. **Next-Generation Firewall (NGFW)** – Advanced firewalls that combine traditional packet filtering with intrusion prevention, deep packet inspection, and threat intelligence.

5. **Host-Based Firewall** – Installed on individual devices to protect them from internal and external threats.

6. **Network-Based Firewall** – Deployed at the network perimeter to protect entire networks from unauthorized access.

sudo -L

```
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:~$
```

TCP traffic on that port to come in

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

```
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:~$
```

Blocking Traffic
sudo iptables -A INPUT -j DROP
sudo iptables -L

```
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:~$
```

Editing iptables
sudo iptables -I INPUT 1 -i lo -j ACCEPTsudo iptables -L

```
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:~$
```

Allowing traffic on ICMP port
sudo iptables -A INPUT -p icmp -j ACCEPT
now list the rules
sudo iptables -L

```
 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:~$
```

Aim : Show the email security pgp using gpg tool between the sender and receiver

Theory :

PGP (Pretty Good Privacy) is an encryption program that provides cryptographic privacy and authentication for data communication. GPG (Gnu Privacy Guard) is a free software replacement for PGP that implements the OpenPGP standard.

PGP using GPG:

1. **Public and Private Key Pair**:
   PGP uses asymmetric encryption, where each user has a **public key** (shared with others) and a **private key** (kept secret). Messages encrypted with the public key can only be decrypted by the private key, and vice versa.

2. **Data Encryption**:
    - **Symmetric Encryption**: For encrypting the actual message, a random symmetric key (session key) is generated. The message is encrypted with this key using a symmetric algorithm like AES.
    - **Asymmetric Encryption**: The session key itself is then encrypted with the recipient's public key using asymmetric encryption (like RSA). This allows the session key to be securely shared.
3. **Digital Signature**:
    - The sender signs the message with their private key, which helps ensure **authenticity** and **integrity**. This digital signature proves the identity of the sender and verifies that the message hasn't been tampered with.
    - The recipient can verify the signature using the sender's public key.
4. **Key Management**:
    - GPG uses **keyrings** to manage the public and private keys. Users can add trusted public keys and use them to encrypt data.
    - GPG also supports **key servers**, where public keys can be shared and retrieved.
5. **Web of Trust**:
    - Unlike centralized systems like SSL, PGP uses a **web of trust** for identity verification. Users sign each other's public keys, building a network of trust without relying on a single central authority.
6. **Message Decryption**:

- When the recipient receives the message, they use their private key to decrypt the symmetric session key. Once they have the session key, they can decrypt the message.

7. **Passphrase Protection**:
    - To enhance security, private keys are often encrypted with a passphrase, adding an extra layer of protection in case the private key is compromised.

Output:

```
shawn@Shawn-Laptop: ~
shawn@Shawn-Laptop: $ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/shawn/.gnupg' created
gpg: keybox '/home/shawn/.gnupg/pubring.kbx' created
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: Shawn
Email address: dcostashawn2004@gmail.com
You selected this USER-ID:
    "Shawn <dcostashawn2004@gmail.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: /home/shawn/.gnupg/trustdb.gpg: trustdb created
gpg: key 5B010A70B6EDA437 marked as ultimately trusted
gpg: directory '/home/shawn/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/shawn/.gnupg/openpgp-revocs.d/AD71B169E4D32C9B12F457905B010A70B6EDA437.rev'
public and secret key created and signed.

pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      AD71B169E4D32C9B12F457905B010A70B6EDA437
uid                      Shawn <dcostashawn2004@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]

shawn@Shawn-Laptop: $
```

```
shawn@Shawn-Laptop: ~    shawn@Shawn-Laptop: ~

Please enter the passphrase to
protect your new key

Passphrase: *********_____________________________

      <OK>                          <Cancel>
```

```
shawn@Shawn-Laptop: ~      shawn@Shawn-Laptop: ~
shawn@Shawn-Laptop:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

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

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

Real name: Vedant
Email address: vedant@gmail.com
You selected this USER-ID:
    "Vedant <vedant@gmail.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: key 483883A43C6AB585 marked as ultimately trusted
gpg: revocation certificate stored as '/home/shawn/.gnupg/openpgp-revocs.d/ABB6FF2770565E62E41926BA483883A43C6AB585.rev'
public and secret key created and signed.

pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      ABB6FF2770565E62E41926BA483883A43C6AB585
uid                      Vedant <vedant@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]

shawn@Shawn-Laptop:~$
```

```
shawn@Shawn-Laptop:~$ gpg --export -a Shawn>demo
shawn@Shawn-Laptop:~$ gpg --export-secret-key -a Shawn>demo_private
```

```
shawn@Shawn-Laptop:~$ gpg --fingerprint vedant@gmail.com
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 2026-10-12
pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      ABB6 FF27 7056 5E62 E419  26BA 4838 83A4 3C6A B585
uid           [ultimate] Vedant <vedant@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]
```

```
shawn@Shawn-Laptop:~$ gpg --import demo
gpg: key 5B010A70B6EDA437: "Shawn <dcostashawn2004@gmail.com>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
```

```
shawn@Shawn-Laptop:~$ gpg --list-keys
/home/shawn/.gnupg/pubring.kbx
-------------------------------
pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      AD71B169E4D32C9B12F457905B010A70B6EDA437
uid           [ultimate] Shawn <dcostashawn2004@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]

pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      ABB6FF2770565E62E41926BA483883A43C6AB585
uid           [ultimate] Vedant <vedant@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]
```

```
shawn@Shawn-Laptop:~$ gpg --sign-key vedant@gmail.com

sec  rsa3072/483883A43C6AB585
     created: 2024-10-12  expires: 2026-10-12  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa3072/2A3F078961DDA25E
     created: 2024-10-12  expires: 2026-10-12  usage: E
[ultimate] (1). Vedant <vedant@gmail.com>


sec  rsa3072/483883A43C6AB585
     created: 2024-10-12  expires: 2026-10-12  usage: SC
     trust: ultimate      validity: ultimate
 Primary key fingerprint: ABB6 FF27 7056 5E62 E419  26BA 4838 83A4 3C6A B585

     Vedant <vedant@gmail.com>

This key is due to expire on 2026-10-12.
Are you sure that you want to sign this key with your
key "Shawn <dcostashawn2004@gmail.com>" (5B010A70B6EDA437)

Really sign? (y/N) y
```

```
shawn@Shawn-Laptop:~$ vi test.txt
shawn@Shawn-Laptop:~$ gpg --encrypt -r vedant@gmail.com test.txt
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 2026-10-12
```

Conclusion: Demonstrated the network security system using open source tools (LO6 is achieved).

Pratham Asnani
T73 63

7/08/25

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Written Assignment 1

Q1 Explain Security Goals in Details with suitable Diagram

Security goals are the fundamental objective of any information security system. These goals ensure that data is protected from threats and available to legitimate users only without compromise

1 Confidentiality
Confidentiality ensures that sensitive information is not disclosed to unauthorized individuals, entities or processes
Objective Prevent data leakage or exposure.

2 Integrity
Integrity means that data is accurate and unaltered during storage, processing or transmission
Objective - Detect and prevent unauthorized modification

3 Availability
Availability ensures that system and data are accessible to authorized users when needed
Objective - Prevent service disruption or downtime

Q2 Define attacks with different types.

An attack is any deliberate or accidental action that undermines the security of information systems by exploiting vulnerabilities

Attacks are broadly categorized into two types

1 Active attack
The attacker modifies disrupts or destroys data or services
Impact- Data alteration, corruption, denial of service.
Example
Masquerade attack.

2 Passive attack
The attacker monitors or intercepts communication without altering it.
Impact- confidentiality is violated
Example
Eveasdropping

3 Insider attacks
Initiated by trusted individuals within the organization
Impact - Often more dangerous due to internal access privileges
Example- Employee copying data to USB

4 Outsider Attacks
Performed by external agent
Example - External Cyberattacks

Q3 Describe Threat and define its types with Appropriate diagram

A threat is any potential event or action that can exploit a system's vulnerability to cause harm, such as data loss, unauthorized access or service unavailability

Threat Categories
1 Natural threat
Threats arising from natural events
Examples - Earthquakes, floods, etc

2 Human Threats
These are threats caused by human actions either intentionally or unintentionally

3 Technical Threats.
Arise due to failures in hardware or software systems
Examples
Software bugs and buffer overflow
Hardware malfunction

4 Logical Threats
Software based threats occurred through code or logic manipulation
Example
Viruses, worms, Trojan horse, SQL injection

5 Physical threat
Physical access or damage to network infrastructure
Example
Theft of devices
Unauthorized entry into server rooms

Pratham Asnani
63 T13

SL Written Assignment - II 11/09/25

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(3)

Intrusion Detection System (IDS)

With the increasing reliance on computer network and internet connectivity, the risk of unauthorized access, cyberattacks and data breaches has grown significantly. An intrusion Detection System (IDS) is a security mechanism designed to detect suspicious activities, unauthorized access, or malicious attacks with a computer system or network. IDS acts like an 'alarm system' for network and computers, alerting administrators when unusual or potentially harmful behavior occurs.

Types of IDE

1 Network based Intrusion Detection System (NIDS)

i Monitors traffic across the entire network
ii Detects attacks such as denial of service (DoS), scanning or unauthorized access attempts.

2 Host-based Intrusion Detection System (HIDS)
i Installed on individual hosts or devices
ii Monitors system log, file integrity and operating system activities.

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Objectives of IDS

1 To detect unauthorized access to computer system

2 To identify potential security breaches in real time.

3 To provide alerts and logs for analysis

4 To support forensic investigation after an attack

5 To complements other security tools.

Working of IDS

The operations of IDS involves several steps

1 Data collection
Captures network packets, system log, and application activity

2 Preprocessing
Organizes and filters collected data

3 Analysis
Detects intrusion using either
i Signature based detection
ii Anomaly based detection

4 Alert Generation
If suspicious activity is detected, IDS generates alerts

5 Response
Alerts are logged and set to administrator

Specialized Types of IDS

1 Hybrid IDS
Combines both signature and anomaly based methods for better accuracy

2 Distributed IDS
Works across multiple network locations to detect coordinated attacks

3 Protocol based IDS
Monitors specific applications protocol (HTTPS, DNS)

Applications of IDS

1 Enterprise Network
To secure sensitive organizational data

2 Banking and financial System
For fraud detection and prevention

3 Government and Defense
To protect confidential and classified information

4 Helthcare System
To safeguard patent records from cyberattacks

5 Education Institutions
To prevent misuse of network by student or outside

Advantages of IDS

1 Provides real time monitoring of network and system activities

2 Detects unauthorized access attempts

3 Helps in forensic analysis by maintaining log of attacks

4 Can identify both external threats and internal misuse

5 Increases network visibility for administrators

Limitations of IDS

1 IDS only detects attacks it cannot block them

2 May generate false positive

3 Consumes resources which can slow down performance in large network.

Conclusion

Intrusion Detection System (IDS) are an essential component of modern cybersecurity infrastructure. They work as a watchdog continuously monitoring and detecting suspicious activities within network and hosts.