Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ AYUSH SANJAY SHARMA

of I.T Department, Semester V with

Roll No. 117 has completed a course of the necessary

experiments in the subject SECURITY LAB under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2025 - 2026

Teacher In- Charge

Head of the Department

Date 09/10/25

Principal

THADOMAL SHAHANI TSEC ENGINEERING SOLLEGE

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Written Assignment covering basic concepts of security like security goals, attacks with different types & threats and define its types with appropriate diagram.	1-5	14/07/25	
2.	Breaking shift ciphes & monoalphabetic substitution ciphes using frequency analysis method	6-8	21/07/25	
3.	Cryptoanalysis or decoding of polyalphabetic ciphers - playfair, vigenere.	9-13	22/07/25	
4.	Block ciphes modes of operation using advanced encryption standard (AES)	14-19	04/08/25	
5.	Implementation and analysis of RSA cryptosystem & digital signature scheme using RSA	20-24	11/08/25	
6.	To explore hashdeep tool in Kali Linux for generating, matching & auditing hash of files	25-28	18/08/25	
7.	Study the use of network reconnaissance tools like whois, dig, Traceroute, nslookup, nikto, dmitry to gather info about networks & domain registers	29-36	25/08/25 ~~01/09/25~~	
8.	Installation of nmap & using it with diff. operations to scan open ports, perform finger printing, ping, TCP port scan, UDP port scan, etc.	37-44	08/09/25	
9.	Simulate DOS attack using HPING 3	45-46	15/09/25	
10.	To study & configure firewalls using IP tables	47-50	22/09/25	
11.	Installing snort, setting in intrusion detection mode & writing rules for intrusion detection	51-57	25/09/25	
12.	Explore the GPG tool of Linux to implement email security	58-61	29/09/25	
13.	Written Assignment 2	62-66	06/10/25	
14.	Class Assignment	67-74	08/10/25	

Rahul 9/10/25

Rahul 9/10/25

AYUSH SHARMA
T22-117

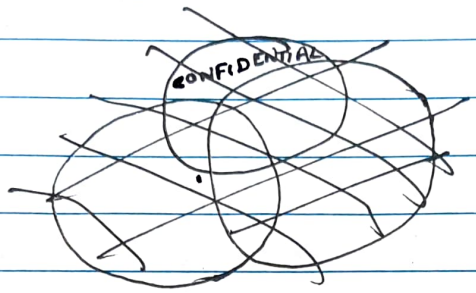
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

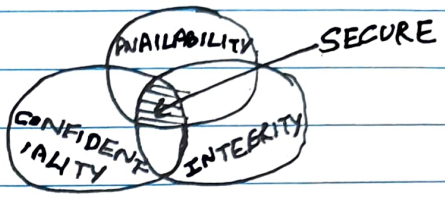
SL WRITTEN ASSIGNMENT-1

06/10/25

Q1. Explain security goals with a ~~se~~ suitable diagram.

Ans





→ The 3 security goals in network security are:

- CONFIDENTIALITY:

→ Confidentiality ensures that data is only accessible to genuine & authorized users. It helps prevent unauthorized access to the data, exploiting the information.
Some methods to ensure confidentiality are:
→ Encrypt raw data using encryption techniques.
→ Use biometric based access control for authorization.
→ Use multi-factor authentication to reduce data exploitation chances.
→ Implement firewalls & IDS to ensure that no third party can access data without permission.

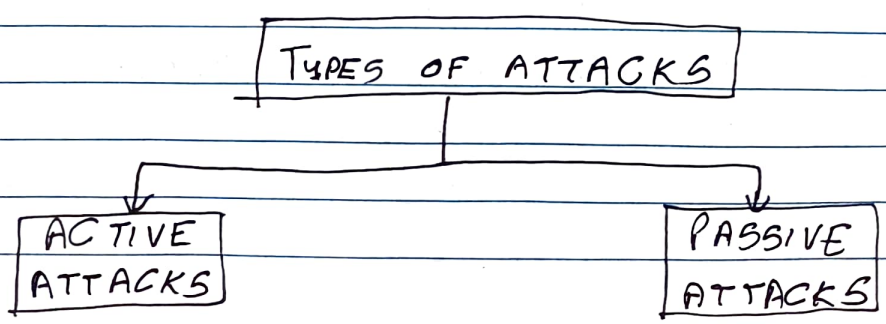
- INTEGRITY

→ Integrity refers to ensuring that the data is in its original form and not altered during the transmission and reaches the end user in its correct form. It provides safeguards from the modification of data by unauthorized users & determines that the source of information is legit.
Some methods to ensure integrity are:
→ Backup the data & resources to ensure its safety during system failure.

→ Employ a version control system to track & analyze any modifications
→ Use access control system to avoid unauthorised modification of data.

• AVAILABILITY
→ It helps delivers the data as and when authorized users require it without errors like denial of service. The data needs to be constantly available for access. It is the guarantee of the reliability of the data.
Some methods used to ensure availability are:
→ Maintain backup servers which can be used in case of system failures.
→ Install firewalls to ensure that the systems doesn't get compromised.
→ Place the backups in geographically isolated places to ~~ensure~~ avoid damages due to any disasters (natural calamity/riots, etc.

Q2. Define attacks with its different types.
Ans An attack is a malicious attempt to gain unauthorized access to digital systems or networks to steal, expose, alter, disable or destroy data, applications or other digital assets.



THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- Active attack:

→ An active attack directly interferes with a system, network or data to disrupt, alter or damage it.
→ It involves a direct interaction with the target.
→ It attempts to alter data content or system functionality.
→ It often leaves evidence making it easier to detect.
- ~~Types of active~~ Examples:
① Denial of service (DOS):- Overwhelms a system with traffic, making it unavailable to legitimate users.
② Ransomware:- Encrypts data & demands a ransom for its decryption.
③ Man in the middle (MitM):- Intercepts & potentially alters communication between 2 parties.
④ Masquerade (impersonation):- The attacker pretends to be an ~~unauthorized~~ trusted entity to gain unauthorized access.

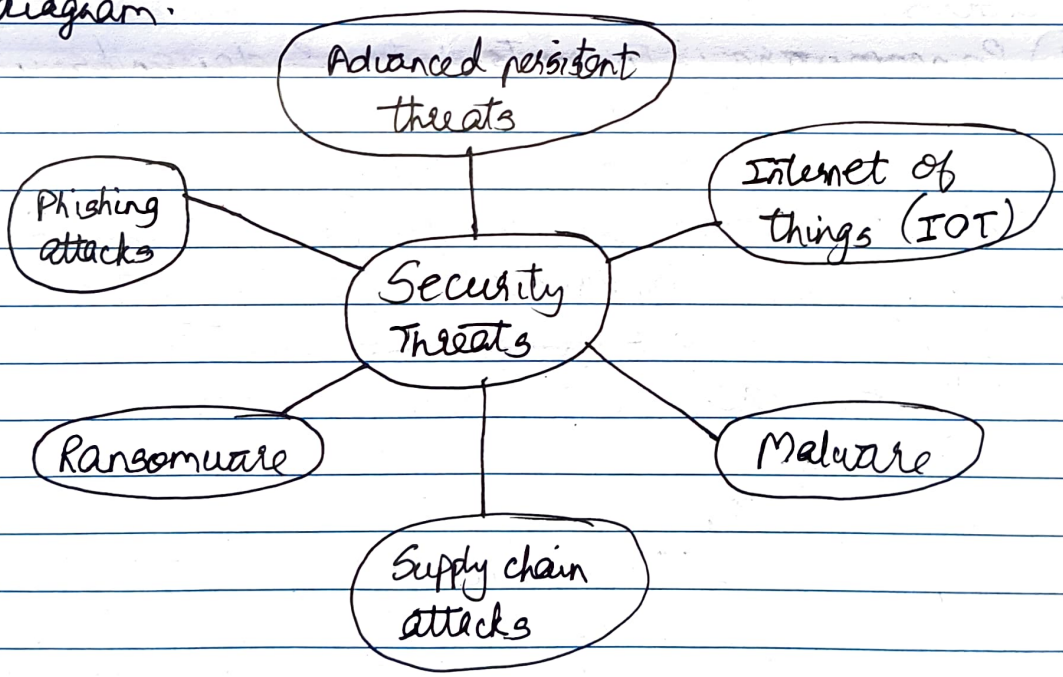
- Passive attack:

→ A passive attack involves an attacker observing or listening to data & systems w/o modifying them, aiming to steal information.
→ The attacker doesn't modify or interact with the target.
→ Difficult to detect because often no direct evidence is left behind.
→ Often used to gather intelligence for future, more targeted attacks.

- Examples:
① Eavesdropping :- Secretly listening to network communications or private calls.
② Traffic analysis :- Monitoring network traffic patterns to infer sensitive information
③ Packet sniffing :- Capturing & analyzing unencrypted data packets to extract information.
④ Footprinting :- Gathering information about a system or network for future exploitation

Q3. Describe threats and define its types with appropriate diagram.

Ans



→ Network security threats are dangers that aim to compromise a network's confidentiality, integrity or availability through actions like unauthorized access, data theft or service disruption
→ Organizations must use robust security measures ~~& educate~~ to safeguard against these threats.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- Common network security threats are:

→ Malware :- A broad category of malicious software (like viruses & spyware) designed to damage systems, steal data or compromise operating systems.

→ Phishing :- Fraudulent attempts to trick individuals into revealing sensitive information often through fake emails or messages that appear to be from legitimate sources.

→ Ransomware :- Malware that locks a user's computer or data, demanding a ransom for its release.

→ Distributed denial of service (DDOS) :- These attacks flood a network with excessive traffic from multiple compromised systems, making it unavailable to legitimate users.

→ Man in the middle (MitM) attacks :- A malicious actor intercepts communication between 2 unsuspecting parties, enabling them to eavesdrop on or alter the data exchanged.

→ SQL injection attacks :- Malicious code is injected into databases via website forms, allowing attackers to gain unauthorized access to or manipulate sensitive data.

→ Social engineering :- Exploiting human psychology to manipulate individuals into performing actions or divulging confidential information that can compromise network security.

ASSIGNMENT 2

AIM:

To understand shift cipher and mono alphabet substitution cipher.

THEORY:

Shift cipher is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

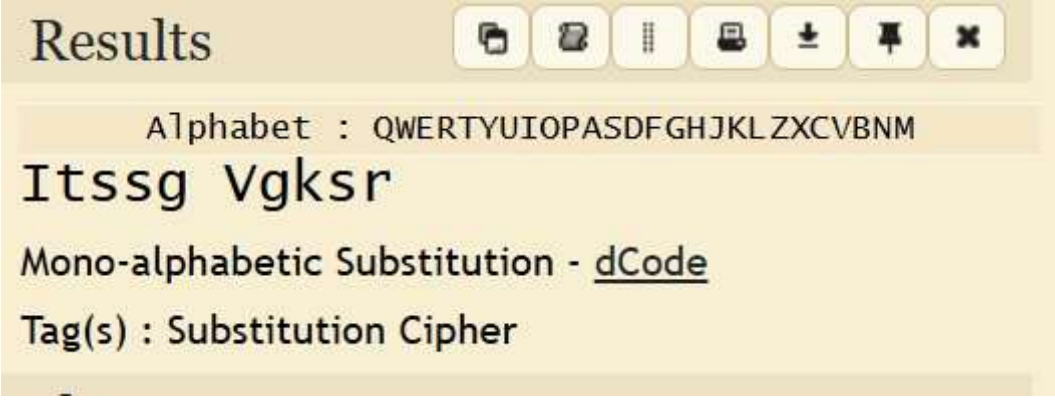
It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on.

The Monoalphabetic Substitution Cipher is also known as the "Simple Substitution Cipher". Monoalphabetic Substitution Ciphers use an individual key mapping function K to replace a specific character α with a character from the mapping K (α). In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.

SCREENSHOTS:

1. SHIFT CYPHER:

2. MONO ALPHABET SUBSTITUTION CYPHER:



CONCLUSION:

Hence, we have implemented and understood the working of a shift and a mono alphabet situation cipher.

LO Mapping: LO-1

ASSIGNMENT 3

AIM:

To understand Playfair and Vigenère cipher.

THEORY:

The Playfair cipher is a manual symmetric encryption technique that encrypts pairs of letters (digraphs) instead of individual letters.

The algorithm consists of 2 steps:

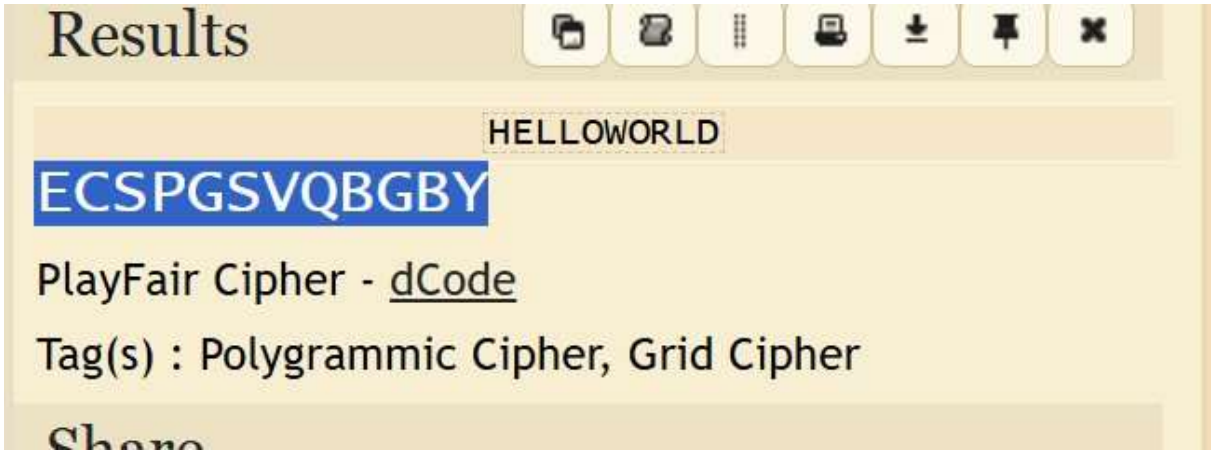
1. Generate the key Square (5x5):
    - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext 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.

Vigenère 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 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.

SCREENSHOTS:

1. PLAYFAIR CIPHER:



PLAYFAIR DECODER

PLAYFAIR CIPHERTEXT

ECSPGSVQBGBY

PLAYFAIR GRID

\	1	2	3	4	5
1	C	I	P	H	E
2	R	A	B	D	F
3	G	K	L	M	N
4	O	Q	S	T	U
5	V	W	X	Y	Z

↕ 5 × ↔ 5 RESIZE

CLEAR

CIPHERABDFGKLMNOQSTUVWXYZ

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

Results

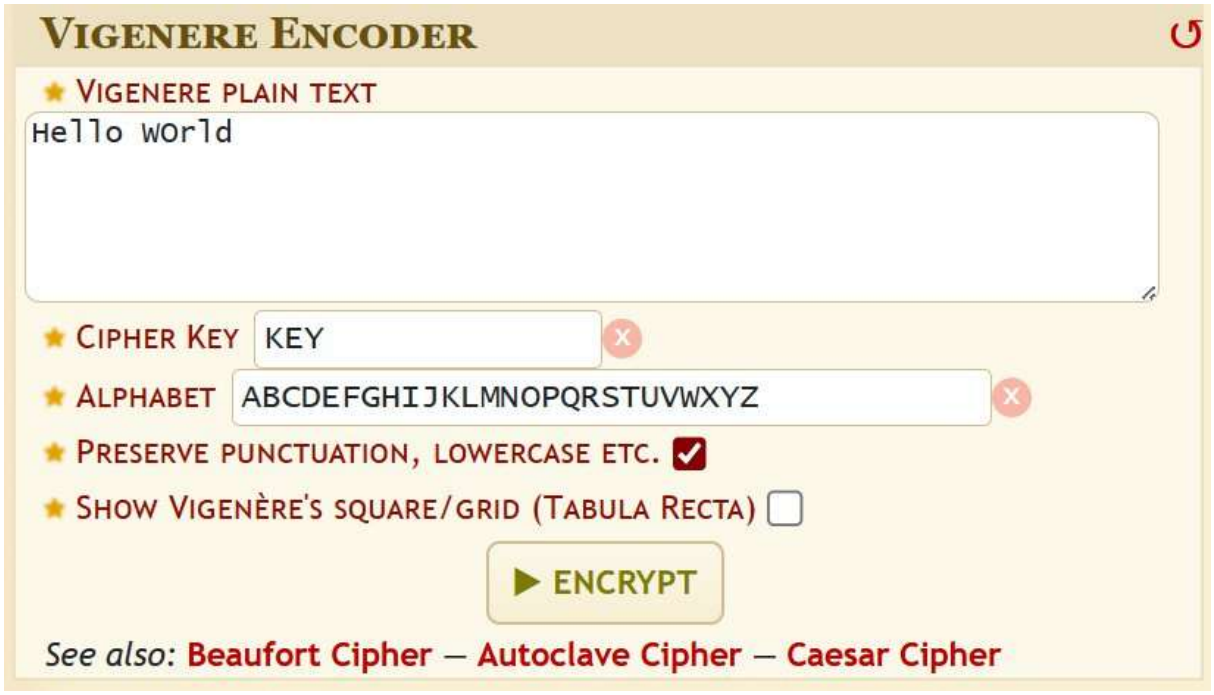
HELXLOWORLDX

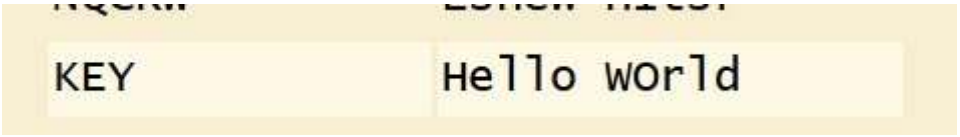
PlayFair Cipher - dCode

Tag(s) : Polygrammic Cipher, Grid Cipher

Share

2. VIGENERE ENCODER:





CONCLUSION:

Hence, we have implemented and understood the working of a Playfair and Vigenère cipher.

LO Mapping: LO-1

ASSIGNMENT 4

AIM:

Study and explain the different types AES modes of operation along with advantages and disadvantages of each mode (EBC, CBC, Counter, cipher feedback and Output Feedback modes).

THEORY:

AES (Advanced Encryption Standard) is a symmetric block cipher that operates on fixed-size blocks (typically 128 bits). Since a block cipher can only encrypt one block at a time, different 'modes of operation' are designed to securely encrypt messages of arbitrary length and improve security. Each mode modifies how encryption is performed on a sequence of blocks, and each has unique properties suitable for different applications.

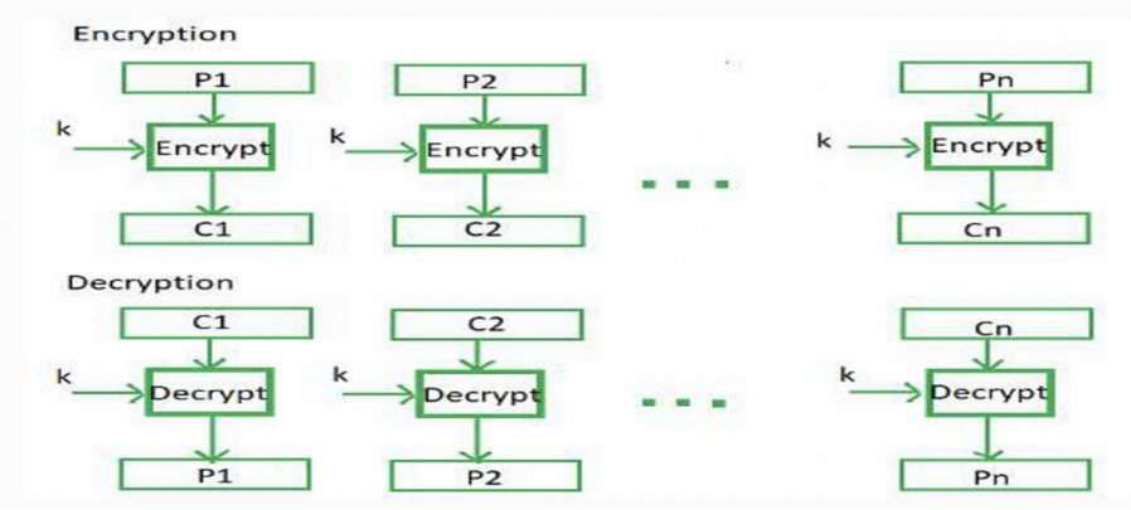
1. Electronic Codebook (ECB) Mode

ECB is the simplest mode of operation. In this mode, the plaintext is divided into equal-sized blocks, and each block is encrypted independently with the same key. This means that identical plaintext blocks always produce identical ciphertext blocks.

Advantages:
- Very simple to implement.
- Supports parallel encryption and decryption of blocks.
- Suitable for small data such as keys.

Disadvantages:
- Not secure for large data because repeating patterns remain visible in ciphertext.
- Identical plaintext blocks yield identical ciphertext blocks.
- Not recommended for encrypting structured data such as images or documents.

Use Cases:
- Encrypting random values like cryptographic keys where patterns are unlikely to repeat.



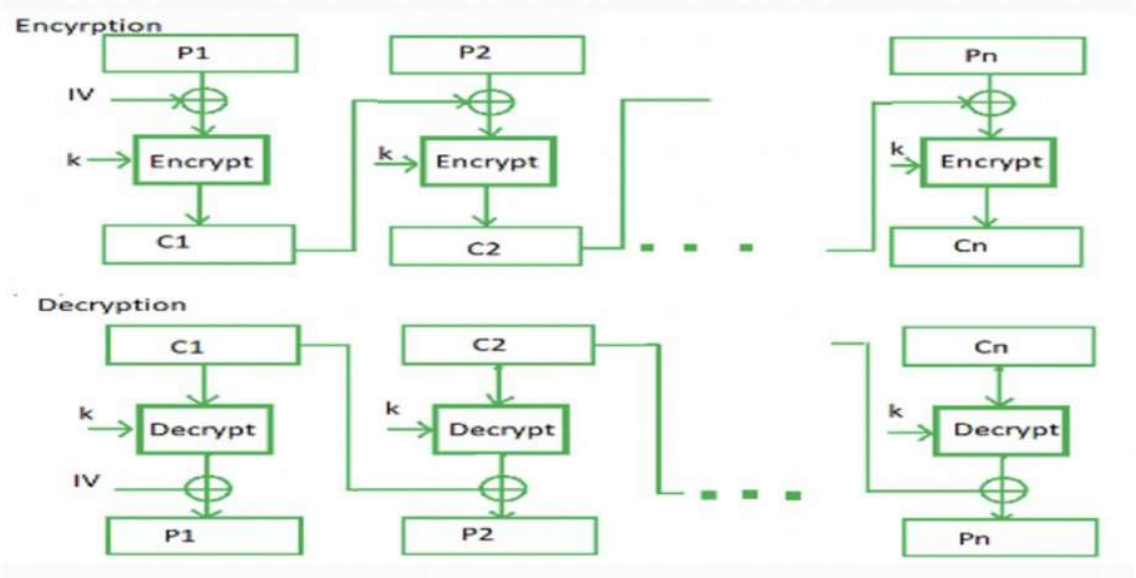
2. Cipher Block Chaining (CBC) Mode

In CBC mode, before encrypting each plaintext block, it is XORed with the previous ciphertext block. An Initialization Vector (IV) is used for the first block to ensure that even identical plaintexts produce different ciphertexts.

Advantages:
- More secure than ECB as it hides data patterns.
- Randomization with IV prevents identical ciphertext outputs for identical plaintexts.
- Widely used in file encryption.

Disadvantages:
- Encryption must be performed sequentially, so parallel encryption is not possible.
- A single bit error in a ciphertext block affects two blocks during decryption.
- Requires proper IV management.

Use Cases:
- Secure file and database encryption where sequential processing is acceptable.



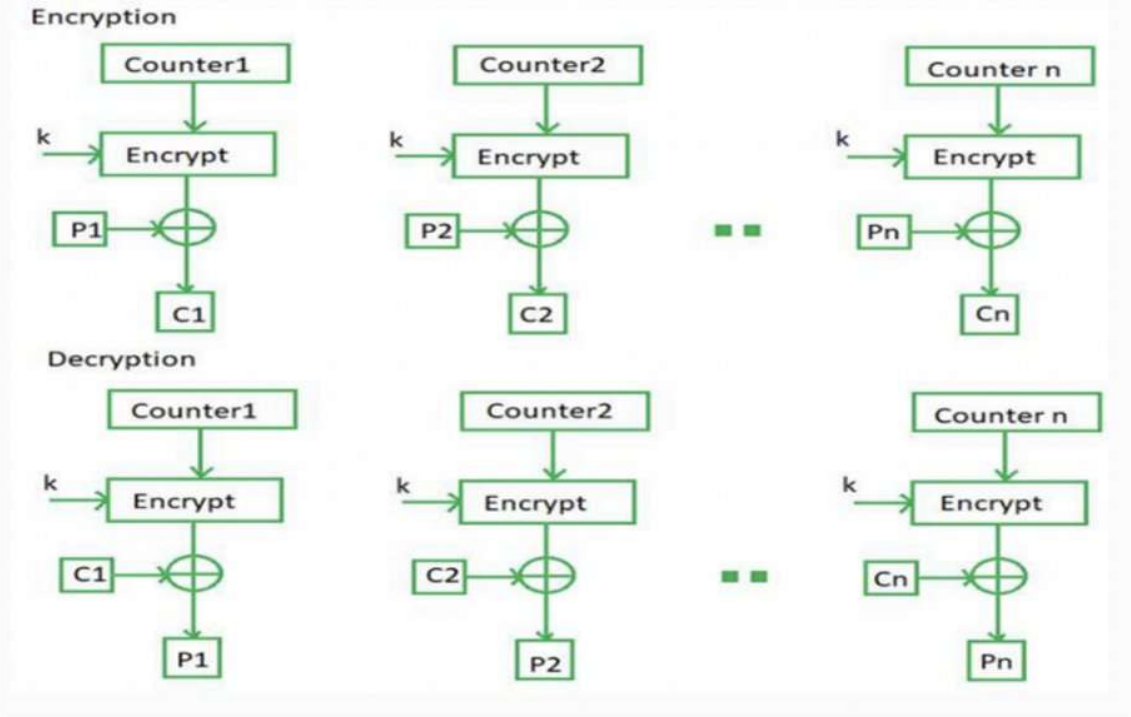
3. Counter (CTR) Mode

CTR mode converts a block cipher into a stream cipher. Instead of encrypting plaintext directly, it encrypts a counter value. The output of the encrypted counter is then XORed with the plaintext to produce ciphertext. The counter is incremented for each block.

Advantages:
- Supports parallel encryption and decryption, making it very efficient.
- Allows random access decryption of any block.
- Provides strong security if counters are unique.

Disadvantages:
- Reuse of counter values with the same key can lead to catastrophic security failures.
- Requires careful synchronization of counter values between sender and receiver.

Use Cases:
- High-speed network encryption such as VPNs.
- Secure data storage systems requiring random access.



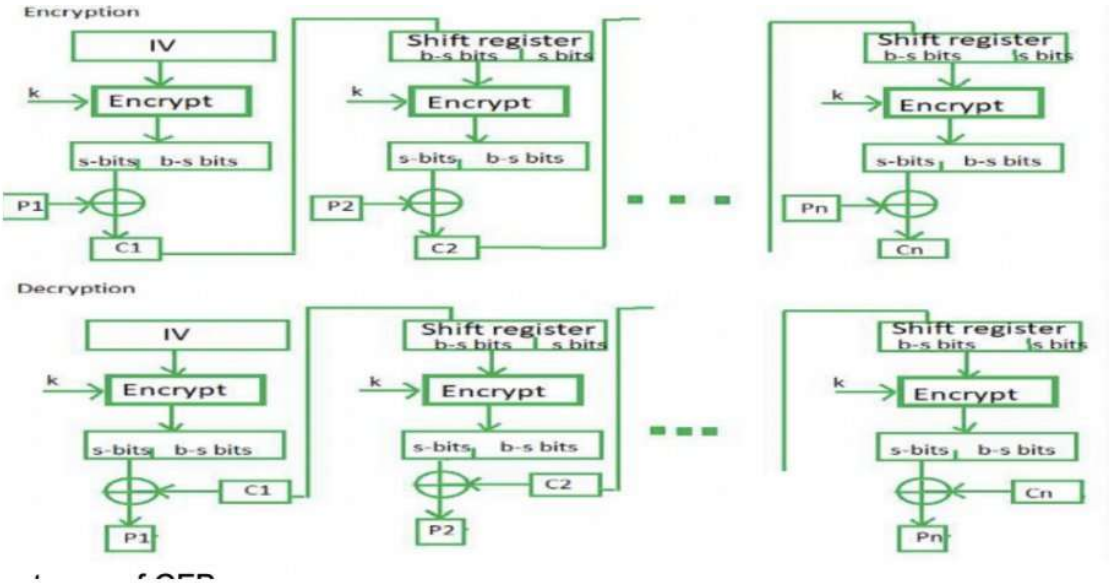
4. Cipher Feedback (CFB) Mode

CFB mode turns a block cipher into a self-synchronizing stream cipher. The previous ciphertext block is encrypted, and the result is XORed with the plaintext to generate the next ciphertext block. It can also operate in smaller units (e.g., 8-bit segments).

Advantages:
- Can be used for real-time encryption of streaming data.
- Self-synchronizing: if a few bits are lost, resynchronization occurs after a few blocks.
- Useful when data arrives in variable sizes.

Disadvantages:
- Encryption must be sequential, preventing parallelism.
- Bit errors in transmission affect more than one block until resynchronization occurs.
- Slower compared to CTR.

Use Cases:
- Encrypted communication channels such as chat applications and streaming media.



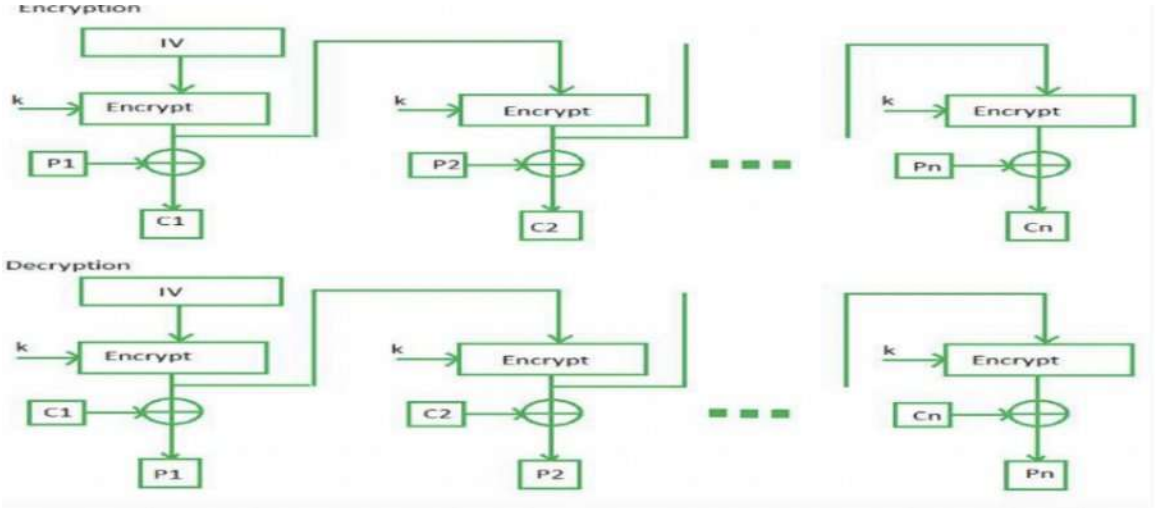
5. Output Feedback (OFB) Mode

OFB mode is similar to CFB, but instead of feeding ciphertext back into the block cipher, it feeds the block cipher output back into itself to generate a keystream. This keystream is XORed with plaintext to produce ciphertext.

Advantages:
- No error propagation: an error in one bit of ciphertext affects only one bit of plaintext.
- Keystream can be precomputed in advance, improving efficiency.
- Suitable for noisy communication channels.

Disadvantages:
- Requires careful IV management; reuse of IV compromises security.
- Cannot provide authentication or integrity checks.
- Not as widely used as CTR.

Use Cases:
- Satellite communication and wireless channels where bit errors are common.



CONCLUSION:

Hence, we have understood the working of different types AES modes of operation along with their advantages and disadvantages.

LO Mapping: LO-2

ASSIGNMENT 5

AIM:

Implementation and analysis of RSA cryptosystem and digital signature scheme using RSA.

THEORY:

RSA Algorithm Overview

RSA is an asymmetric cryptographic algorithm that uses a pair of keys:

• Public Key: Used for encryption, consists of modulus (n) and public exponent (e)

• Private Key: Used for decryption, consists of modulus (n) and private exponent (d)

Key Generation

1. Choose two large prime numbers p and q

2. Compute n = p × q (the modulus)

3. Compute φ(n) = (p-1) × (q-1) (Euler's totient function)

4. Choose e such that 1 < e < φ(n) and gcd(e, φ(n)) = 1

5. Compute d ≡ e ¹ (mod φ(n)) (the private exponent)

RSA-Based Digital Signatures

RSA (Rivest-Shamir-Adleman) is widely used for digital signatures due to its mathematical

foundation:

Key Properties:

• Key Generation: Create a public-private key pair (e,n) and (d,n)

• Signing: Encrypt the message hash with the private key: s=H(m)dmodn

• Verification: Decrypt the signature with the public key: H(m)=?semodn

RSA Security Assumption: The security of RSA digital signatures relies on the difficulty of the

Integer Factorization Problem: Given n=p×q, find p and q

Trapdoor Function: RSA uses a trapdoor one-way function where:

• Easy direction: Computing y=xemodn is efficient

• Hard direction: Computing x=ydmodn without knowing d is computationally infeasible

Mathematical Representation

For RSA digital signatures:

Signature Generation: s=H(m)dmodn

Signature Verification: H(m)=semodn

Where:

• H(m) is the hash of message m

• d is the private key exponent

• e is the public key exponent

• n is the modulus ($n=p\times q$ for primes p and q)

RSA Key Generation Process:

1. Choose two large prime numbers p and q

2. Compute $n=p\times q$

3. Compute Euler's totient function: $\phi(n)=(p-1)(q-1)$

4. Choose public exponent e such that $1<e<\phi(n)$ and $\gcd(e,\phi(n))=1$

5. Compute private exponent d such that $d\equiv e-1(\mathrm{mod}\phi(n))$

Digital Signature Algorithm Steps:

Signing Process:

1. Compute message hash: $h=H(m)$

2. Generate signature: $s=h^d \mathrm{mod} n$

3. Send (m,s) to verifier

Verification Process:

1. Compute message hash: $h'=H(m)$

2. Decrypt signature: $h''=s^e \mathrm{mod} n$

3. Verify: signature is valid if $h'=h''$

Step 1: Enter Plaintext and Generate Hash

Plaintext (string): hi how are you — Generate SHA-1 Hash

Hash output (hex): 4b7f17909a25c098b51f4b6bb38d6ac56e8acc49

Step 2: Input Hash to RSA

Copy the hash value above to the input field below:

Input to RSA (hex): 4b7f17909a25c098b51f4b6bb38d6ac56e8acc49 — Apply RSA Signature

Step 3: View Digital Signature Results

Copy the hash value above to the input field below:

Input to RSA (hex): 4b7f17909a25c098b51f4b6bb38d6ac56e8acc49 — Apply RSA Signature

Step 3: View Digital Signature Results

Step 4: Select RSA Public Key

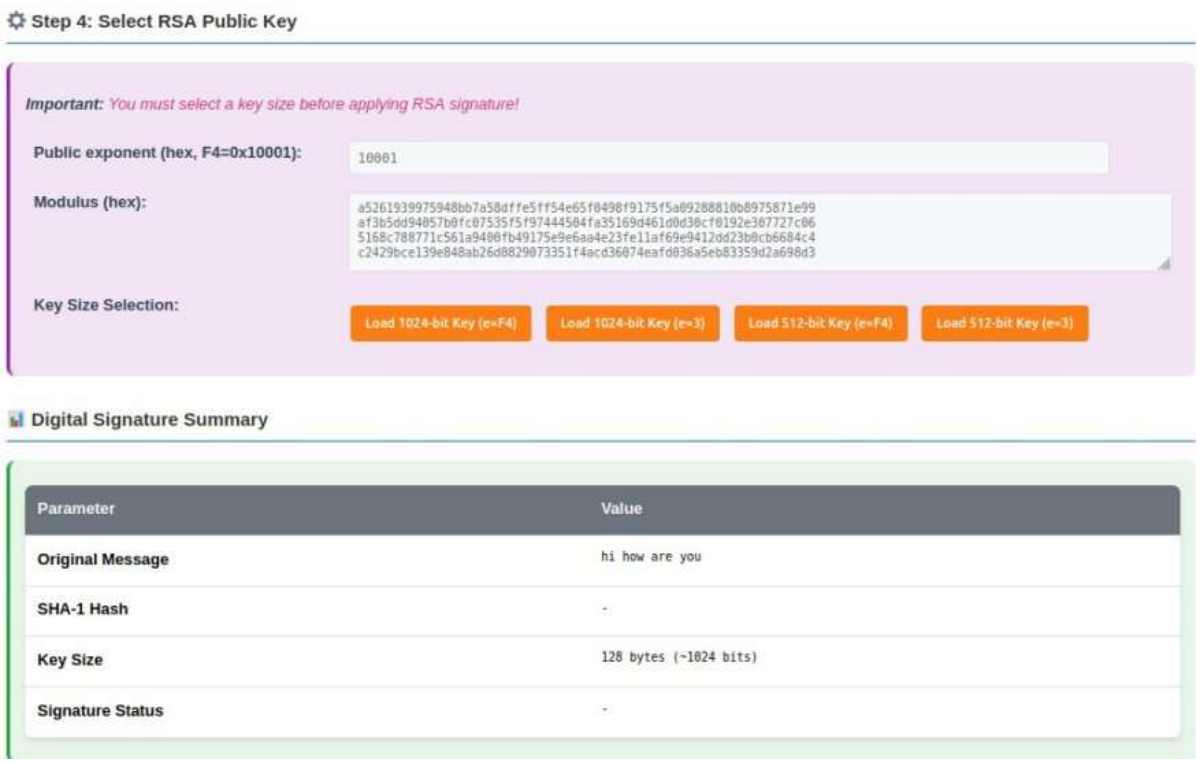
Important: *You must select a key size before applying RSA signature!*

Public exponent (hex, F4=0x10001): 10001

Modulus (hex):

```
a5261939975948bb7a58dffe5ff54e65f0498f9175f5a09288810b8975871e99
af3b5dd94057b0fc07535f5f97444504fa35169d461d0d30cf0192e307727c06
5168c788771c561a9400fb49175e9e6aa4e23fe11af69e9412dd23b0cb6684c4
c2429bce139e848ab26d0829073351f4acd36074eafd036a5eb83359d2a698d3
```

Key Size Selection: Load 1024-bit Key (e=F4) — Load 1024-bit Key (e=3) — Load 512-bit Key (e=F4) — Load 512-bit Key (e=3)



CONCLUSION:

Hence, we have understood and implemented RSA cryptosystem.

LO Mapping: LO-2

ASSIGNMENT 6

AIM:

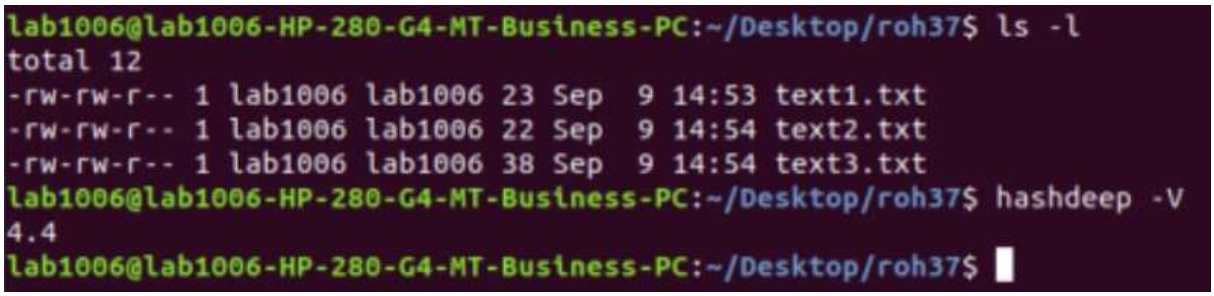
To explore hashdeep tool in Kali Linux for generating, matching and auditing hash of files.

THEORY:

Hashdeep is a command-line tool in Linux used to compute and verify cryptographic hashes like MD5, SHA1, and SHA256 across files and directories. It supports recursive scanning, making it ideal for checking large datasets or backups for integrity. You can generate hashsets, compare them later, and detect any tampering or corruption with high precision.

It's especially useful in forensic analysis, system audits, and secure file verification workflows. Hashdeep's audit mode allows you to compare current files against a known hashset, flagging mismatches or missing files. A typical command looks like: hashdeep -r -c md5,sha1,sha256 -l -o t folder/ , which recursively hashes files and outputs a timestamped log.

SCREENSHOTS:



```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -h
hashdeep version 4.4 by Jesse Kornblum and Simson Garfinkel.
$ hashdeep [OPTION]... [FILES]...
-c <alg1,[alg2]> - Compute hashes only. Defaults are MD5 and SHA-256
                  legal values: md5,sha1,sha256,tiger,whirlpool,
-p <size> - piecewise mode. Files are broken into blocks for hashing
-r        - recursive mode. All subdirectories are traversed
-d        - output in DFXML (Digital Forensics XML)
-k <file> - add a file of known hashes
-a        - audit mode. Validates FILES against known hashes. Requires -k
-m        - matching mode. Requires -k
-x        - negative matching mode. Requires -k
-w        - in -m mode, displays which known file was matched
-M and -X act like -m and -x, but display hashes of matching files
-e        - compute estimated time remaining for each file
-s        - silent mode. Suppress all error messages
-b        - prints only the bare name of files; all path information is omitted
-l        - print relative paths for filenames
-i/-I     - only process files smaller than the given threshold
-o        - only process certain types of files. See README/manpage
-v        - verbose mode. Use again to be more verbose
-d        - output in DFXML; -W FILE - write to FILE.
-j <num>  - use num threads (default 6)
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -b text1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -b text1.txt
##
23,6199d58ee08b958185d74ef62f7125e2,78f295acc6af8fa286c6241792878b41782198 5e7734c19876f7db39af1eda3f,text1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep text1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep text1.txt
##
23,6199d58ee08b958185d74ef62f7125e2,78f295acc6af8fa286c6241792878b41782198 5e7734c19876f7db39af1eda3f,/home/lab1006/Desktop/roh37/text1.txt
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -b text1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -b text1.txt
##
23,6199d58ee08b958185d74ef62f7125e2,78f295acc6af8fa286c6241792878b41782198 5e7734c19876f7db39af1eda3f,text1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep text1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep text1.txt
##
23,6199d58ee08b958185d74ef62f7125e2,78f295acc6af8fa286c6241792878b41782198 5e7734c19876f7db39af1eda3f,/home/lab1006/Desktop/roh37/text1.txt
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c md5 *txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -c md5 text1.txt text2.txt text3.txt
##
23,6199d58ee00b958185d74ef62f7125e2,/home/lab1006/Desktop/roh37/text1.txt
22,0ac454bc9f8df80b34f7d07ec7cc69c9,/home/lab1006/Desktop/roh37/text2.txt
38,c4d5c2cda23051b280b2597b8cc332d5,/home/lab1006/Desktop/roh37/text3.txt
```

```
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c md5,sha1 *txt
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -c md5,sha1 text1.txt text2.txt text3.txt
##
38,c4d5c2cda23051b280b2597b8cc332d5,36f5d4aa7a1dc29fd77d597d9087ecf5e9de74fd,/home/lab1006/Desktop/roh37/text3.txt
23,6199d58ee00b958185d74ef62f7125e2,77de4842b23ff7ab49bfedfffd93dd19ca97e5b8,/home/lab1006/Desktop/roh37/text1.txt
22,0ac454bc9f8df80b34f7d07ec7cc69c9,8eabbff7f3e5e5d27380a5910cb7f047fd750e1b,/home/lab1006/Desktop/roh37/text2.txt
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c md5 -p 10 text3.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -c md5 -p 10 text3.txt
##
10,ff743da0bb4f0c4e99419a6afa2c5f77,/home/lab1006/Desktop/roh37/text3.txt offset 0-9
10,08e82b34c7caeb602dc932c184032858,/home/lab1006/Desktop/roh37/text3.txt offset 10-19
10,0d624e01be47b42c9af2e81492df5ab8,/home/lab1006/Desktop/roh37/text3.txt offset 20-29
8,27c625471ca47887ec79715b82eb862c,/home/lab1006/Desktop/roh37/text3.txt offset 30-37
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c  md5 -r /home/lab1006/Desktop/roh37
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -c md5 -r /home/lab1006/Desktop/roh37
##
22,0ac454bc9f8df80b34f7d07ec7cc69c9,/home/lab1006/Desktop/roh37/text2.txt
23,6199d58ee00b958185d74ef62f7125e2,/home/lab1006/Desktop/roh37/text1.txt
38,c4d5c2cda23051b280b2597b8cc332d5,/home/lab1006/Desktop/roh37/text3.txt
```

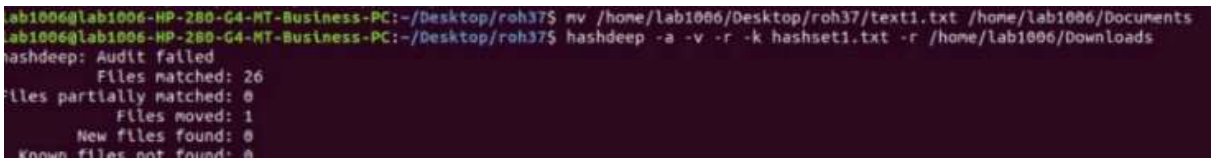
```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c  md5 -r /home/lab1006/Desktop/roh37
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006/Desktop/roh37
## $ hashdeep -c md5 -r /home/lab1006/Desktop/roh37
##
22,0ac454bc9f8df80b34f7d07ec7cc69c9,/home/lab1006/Desktop/roh37/text2.txt
23,6199d58ee00b958185d74ef62f7125e2,/home/lab1006/Desktop/roh37/text1.txt
38,c4d5c2cda23051b280b2597b8cc332d5,/home/lab1006/Desktop/roh37/text3.txt
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ md5deep -m   hashset1.txt *
/home/lab1006/Desktop/roh37/text1.txt
/home/lab1006/Desktop/roh37/text3.txt
/home/lab1006/Desktop/roh37/text2.txt
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ md5deep -s -x hashset1.txt *
/home/lab1006/Desktop/roh37/hashset1.txt
/home/lab1006/Desktop/roh37/unh.txt
```

```
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -c md5,sha1,sha256 -r /home/lab1006/Downloads>hashset1.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -a -r -k hashset1.txt -r /home/lab1006/Downloads
ashdeep: Audit passed
```

```
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ touch /home/lab1006/Downloads/newfile.txt
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$ hashdeep -a -r -k hashset1.txt -r /home/lab1006/Downloads
ashdeep: Audit failed
ab1006@lab1006-HP-280-G4-MT-Business-PC:~/Desktop/roh37$
```



CONCLUSION:

Hence, we have understood and implemented Hashdeep.

LO Mapping: LO-2

ASSIGNMENT 7

AIM:

Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup, nikto, dmitry to gather information about network and domain registrars.

THEORY:

1. WHOIS: The whois command queries public WHOIS databases to get information about a domain name or IP address, such as the registrar, registration/expiry dates, and name servers. It's commonly used for domain ownership checks, network troubleshooting, and cybersecurity investigations.
2. traceroute: The whois command queries public WHOIS databases to get information about a domain name or IP address, such as the registrar, registration/expiry dates, and name servers. It's commonly used for domain ownership checks, network troubleshooting, and cybersecurity investigations.
3. nikto: Nikto is an open-source web server scanner that checks a target for vulnerabilities, misconfigurations, outdated software, and dangerous files. It's commonly used in penetration testing to quickly identify potential security risks in web applications and servers.
4. dmitry: DMitry (Deepmagic Information Gathering Tool) is a command-line tool used for gathering information about a host, such as WHOIS data, subdomains, email addresses, and open ports. It's often used in reconnaissance during penetration testing or security assessments.
5. nslookup: The nslookup command queries DNS servers to get information about a domain name or IP address, such as its corresponding IP, mail server records, or other DNS entries. It's mainly used for DNS troubleshooting and verifying domain name resolution.
6. dig: The dig (Domain Information Groper) command queries DNS servers to retrieve detailed DNS records for a domain, such as A, MX, TXT, and NS records. It's widely used for DNS troubleshooting because it provides more flexibility and detailed output than nslookup.

SCREENSHOTS:

```
In Europe, at +44.02032062220
--
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ whois 64.233.170.139

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2025, American Registry for Internet Numbers, Ltd.
#


NetRange:       64.233.160.0 - 64.233.191.255
CIDR:           64.233.160.0/19
NetName:        GOOGLE
NetHandle:      NET-64-233-160-0-1
Parent:         NET64 (NET-64-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Google LLC (GOGL)
RegDate:        2003-08-18
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute google.com
traceroute to google.com (142.251.43.14), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.463 ms  0.441 ms  0.564 ms
 2  203.212.25.1 (203.212.25.1)  1.718 ms  2.471 ms  2.651 ms
 3  203.212.24.53 (203.212.24.53)  2.457 ms  7.323 ms  2.626 ms
 4  53.177.100.175.mipl.com (175.100.177.53)  4.674 ms  4.456 ms  4.575 ms
 5  * * *
 6  22.188.100.175.mipl.com (175.100.188.22)  4.219 ms  2.683 ms  2.698 ms
 7  * * *
 8  142.250.214.110 (142.250.214.110)  2.602 ms 172.253.77.20 (172.253.77.20)  2
.581 ms 142.250.238.198 (142.250.238.198)  3.459 ms
 9  192.178.110.104 (192.178.110.104)  4.734 ms 192.178.110.204 (192.178.110.204
)  3.236 ms 192.178.110.104 (192.178.110.104)  3.839 ms
10  tsa03s08-in-f14.1e100.net (142.251.43.14)  3.711 ms  3.389 ms  3.173 ms
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute tsec.edu -q 3
traceroute to tsec.edu (50.6.173.12), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.607 ms  0.573 ms  0.490 ms
 2  203.212.25.1 (203.212.25.1)  2.312 ms  2.251 ms  2.279 ms
 3  203.212.24.53 (203.212.24.53)  2.199 ms  2.248 ms  2.265 ms
 4  53.177.100.175.mipl.com (175.100.177.53)  3.292 ms  3.343 ms  3.344 ms
 5  172.16.2.101 (172.16.2.101)  3.586 ms  4.322 ms  3.942 ms
 6  121.241.42.57.static-mumbai.vsnl.net.in (121.241.43.57)  3.954 ms  2.901 ms
 2.807 ms
 7  * * *
 8  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net (180.87.38.5)  3.600 ms *  7.660 ms
 9  * if-ae-32-2.tcore2.mlv-mumbai.as6453.net (180.87.38.147)  187.200 ms if-bun
dle-13-2.qcore1.mlv-mumbai.as6453.net (180.87.38.29)  192.065 ms
10  if-bundle-12-2.qcore4.ldn-london.as6453.net (180.87.39.21)  192.981 ms * *
11  * 195.219.213.138 (195.219.213.138)  191.695 ms *
12  * * *
13  be2871.ccr42.lon13.atlas.cogentco.com (154.54.58.185)  126.917 ms be2868.ccr
41.lon13.atlas.cogentco.com (154.54.57.153)  138.116 ms  138.112 ms
14  be2490.ccr42.jfk02.atlas.cogentco.com (154.54.42.85)  203.738 ms  199.931 ms
 be2317.ccr41.jfk02.atlas.cogentco.com (154.54.30.185)  200.402 ms
15  port-channel4188.ccr92.dca04.atlas.cogentco.com (154.54.30.121)  198.681 ms
port-channel8073.ccr91.dca04.atlas.cogentco.com (154.54.170.69)  207.320 ms port
-channel4188.ccr92.dca04.atlas.cogentco.com (154.54.30.121)  197.248 ms
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute -f 2 google.com
traceroute to google.com (142.251.43.14), 30 hops max, 60 byte packets
 2  203.212.25.1 (203.212.25.1)  1.281 ms  1.844 ms  1.816 ms
 3  203.212.24.53 (203.212.24.53)  2.226 ms  2.157 ms  2.227 ms
 4  53.177.100.175.mipl.com (175.100.177.53)  2.574 ms  2.947 ms  3.043 ms
 5  * * 172.16.2.202 (172.16.2.202)  9.801 ms
 6  22.188.100.175.mipl.com (175.100.188.22)  2.775 ms  3.277 ms  2.861 ms
 7  * * *
 8  72.14.233.58 (72.14.233.58)  2.213 ms 192.178.86.240 (192.178.86.240)  3.993
 ms 192.178.86.202 (192.178.86.202)  3.587 ms
 9  192.178.110.106 (192.178.110.106)  3.680 ms 192.178.110.198 (192.178.110.198
)  2.939 ms 142.251.77.99 (142.251.77.99)  2.822 ms
10  142.251.77.69 (142.251.77.69)  3.504 ms 192.178.110.245 (192.178.110.245)  3
.508 ms  3.513 ms
11  142.251.77.99 (142.251.77.99)  3.162 ms pnbomb-bo-in-f14.1e100.net (142.251.
43.14)  3.157 ms  2.700 ms
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h gmail.com
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          142.250.70.37
+ Target Hostname:    gmail.com
+ Target Port:        80
+ Start Time:         2025-07-25 11:25:33 (GMT5.5)
---------------------------------------------------------------------------
+ Server: sffe
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'x-xss-protection' found, with contents: 0
+ Uncommon header 'cross-origin-resource-policy' found, with contents: cross-ori
gin
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ Root page / redirects to: https://mail.google.com/mail/u/0/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Uncommon header 'cross-origin-opener-policy-report-only' found, with contents:
 same-origin; report-to="static-on-bigtable"
+ Uncommon header 'report-to' found, with contents: {"group":"static-on-bigtable
","max_age":2592000,"endpoints":[{"url":"https://csp.withgoogle.com/csp/report-t
o/static-on-bigtable"}]}
+ File/dir '/' in robots.txt returned a non-forbidden or redirect HTTP code (301
)
+ "robots.txt" contains 4 entries which should be manually viewed.
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h example.com -maxtime 3600
- Nikto v2.1.5
---------------------------------------------------------------------------
+ ERROR: Host maximum execution time of 3600 seconds reached
+ Target IP:          96.7.128.175
+ Target Hostname:    example.com
+ Target Port:        80
+ Start Time:         2025-07-25 11:27:25 (GMT5.5)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ Server leaks inodes via ETags, header found with file /, fields: 0x84238dfc809
2e5d9c0dac8ef93371a07:1736799080.121134
+ The anti-clickjacking X-Frame-Options header is not present.
+ Server banner has changed from '' to 'AkamaiNetStorage' which may suggest a WA
F, load balancer or proxy is in place
```

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

Writing output to 'somefile.txt'

HostIP:23.215.0.136
HostName:example.com

Gathered Inet-whois information for 23.215.0.136
---------------------------------


inetnum:        23.111.248.0 - 23.239.127.255
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
```

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

HostIP:142.251.43.14
HostName:google.com

Gathered Inic-whois information for google.com
---------------------------------
   Domain Name: GOOGLE.COM
   Registry Domain ID: 2138514_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteP
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTrans
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateP
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteP
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTrans
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateP
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/w
>>> Last update of whois database: 2025-07-25T06:04:12Z <<<
```

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

Non-authoritative answer:
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns3.google.com.

Authoritative answers can be found from:
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nslookup -timeout=150 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
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nslookup example.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   example.com
Address: 23.192.228.84
Name:   example.com
Address: 23.215.0.138
Name:   example.com
Address: 23.215.0.136
Name:   example.com
Address: 96.7.128.175
Name:   example.com
Address: 96.7.128.198
Name:   example.com
Address: 23.192.228.80
Name:   example.com
Address: 2600:1406:3a00:21::173e:2e66
Name:   example.com
Address: 2600:1406:bc00:53::b81e:94ce
Name:   example.com
```

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

; <<>> DiG 9.11.3-1ubuntu1.18-Ubuntu <<>> example.com +trace
;; global options: +cmd
;; Received 51 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dig @8.8.8.8 gmail.com

; <<>> DiG 9.11.3-1ubuntu1.18-Ubuntu <<>> @8.8.8.8 gmail.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26526
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;gmail.com.                     IN      A

;; ANSWER SECTION:
gmail.com.              300     IN      A       142.250.192.5

;; Query time: 7 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Jul 25 11:36:03 IST 2025
;; MSG SIZE  rcvd: 54
```

CONCLUSION:

Hence, we have understood network reconnaissance and its use and purpose.

LO Mapping: LO-3

ASSIGNMENT 8

AIM:

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

THEORY:

Nmap (Network Mapper) is an open-source security scanner used to discover hosts and services on a computer network. It probes targets and reports what it finds — live hosts, open ports, running services and versions, OS details, firewalls, and more. It's a fundamental tool for network inventory, vulnerability assessment, and penetration testing.

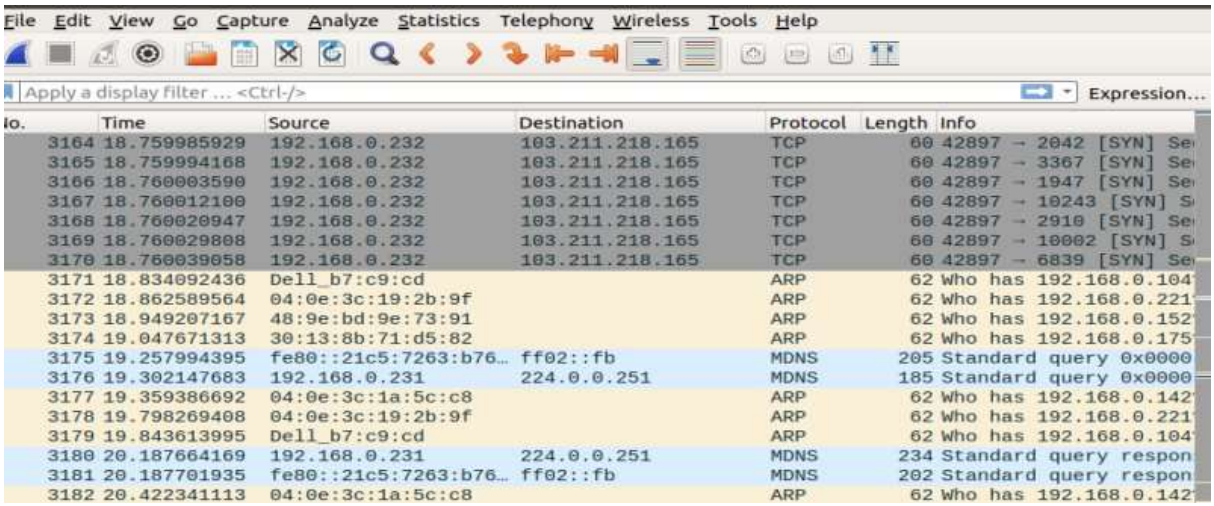
SCREENSHOTS:

1)tcp sync scan

sudo nmap -sS tsec.edu

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since it never completes TCP connections

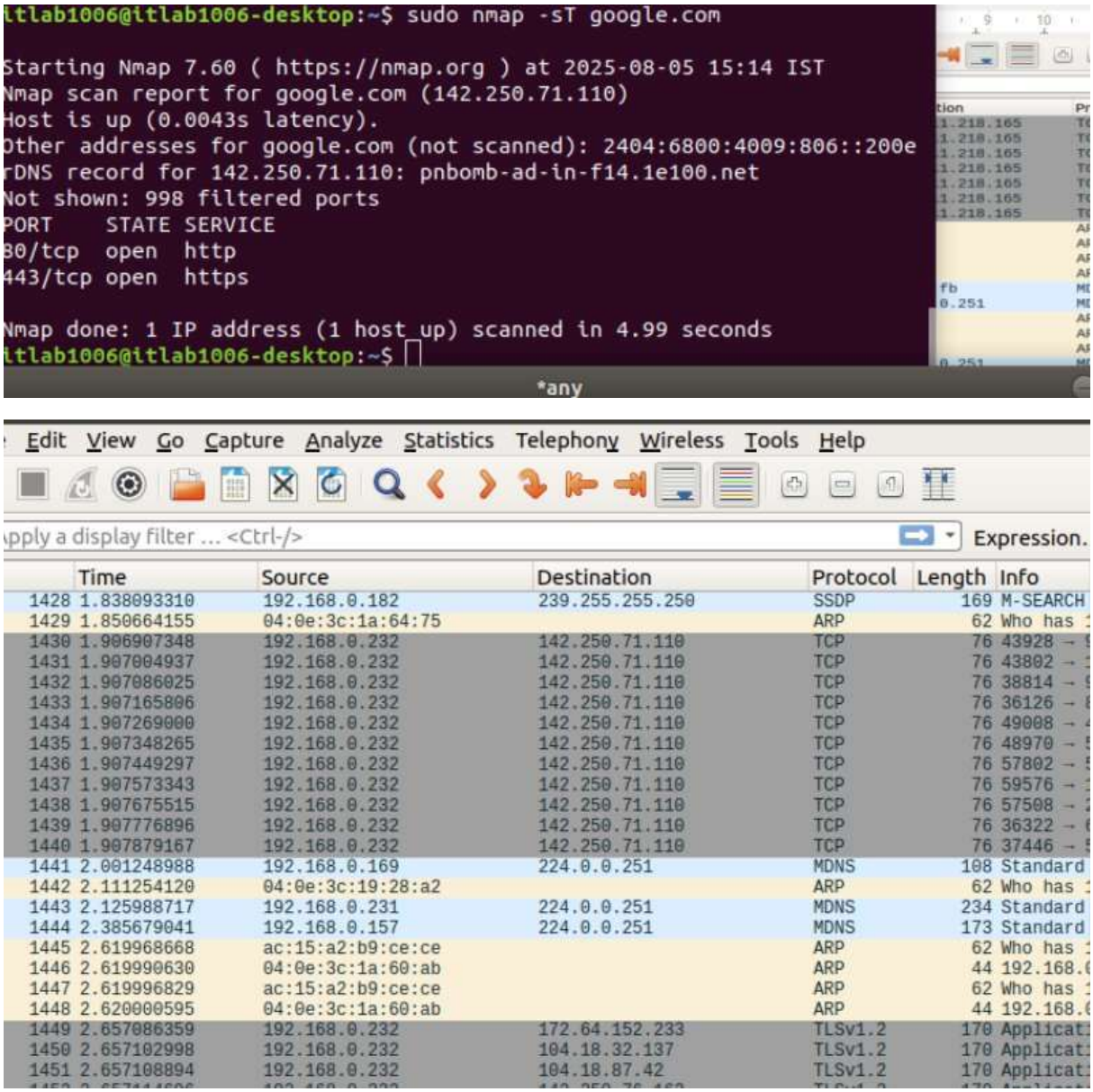
```
ttlab1006@ttlab1006-desktop:~$ sudo nmap -sS tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:05 IST
Nmap scan report for tsec.edu (103.211.218.165)
Host is up (0.042s latency).
rDNS record for 103.211.218.165: vps.tec-sense.com
Not shown: 928 filtered ports, 60 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
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
```



2)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. 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.

Sudo nmap -sT google.com



3)Udp Scan

UDP scan works by sending a UDP packet to every targeted port. For some common ports such as 53 and 161, a protocol-specific payload is sent to increase response rate, but for most ports the packet is empty unless the --data, --data-string, or --data-length options are specified.
Sudo nmap -sU google.com

```
itlab1006@itlab1006-desktop:~$ sudo nmap -sU google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:19 IST
Nmap scan report for google.com (142.250.71.110)
Host is up (0.0039s latency).
Other addresses for google.com (not scanned): 2404:6800:4009:806::200e
rDNS record for 142.250.71.110: pnbomb-ad-in-f14.1e100.net
Not shown: 999 open|filtered ports
PORT      STATE  SERVICE
33459/udp closed unknown
```

4)Null, Fin and Xmas Scan

These three scan types exploit a subtle loophole in the TCP RFC to differentiate between open and closed ports

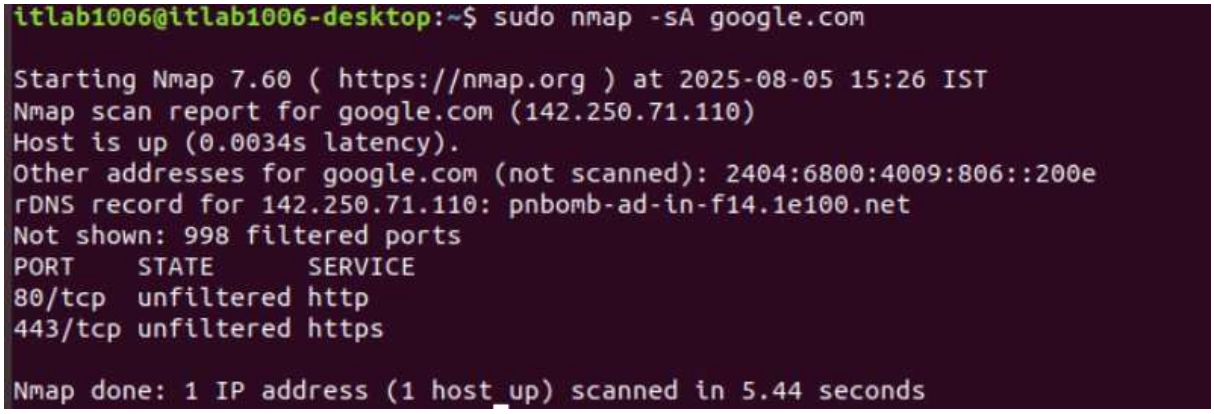
Null scan (-sN) Does not set any bits (TCP flag header is 0)

FIN scan (-sF) Sets just the TCP FIN bit.

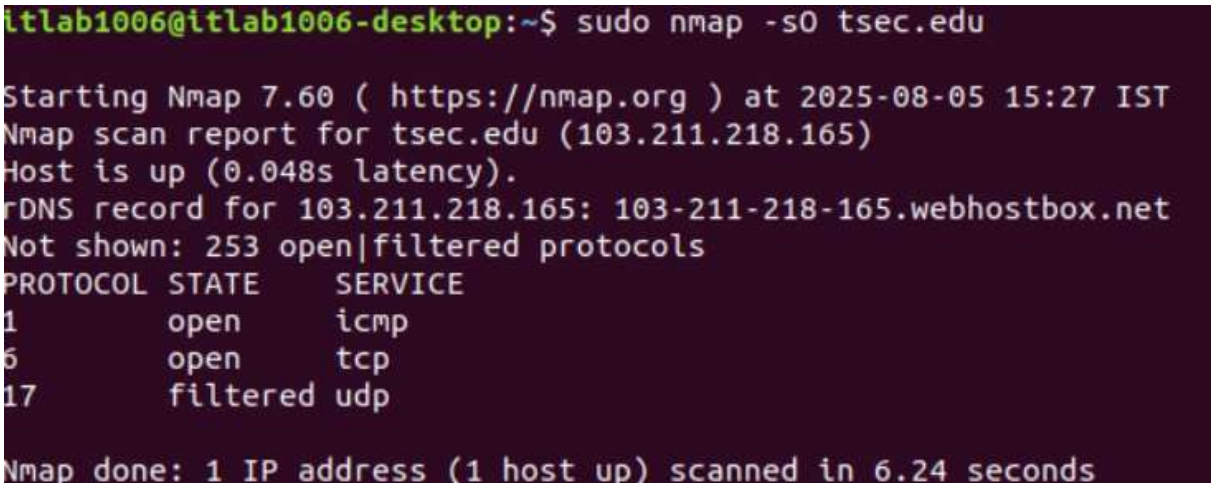
Xmas scan (-sX) Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

```
itlab1006@itlab1006-desktop:~$ sudo nmap -sN tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:23 IST
Nmap scan report for tsec.edu (103.211.218.165)
Host is up (0.031s latency).
rDNS record for 103.211.218.165: 103-211-218-165.webhostbox.net
All 1000 scanned ports on tsec.edu (103.211.218.165) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 5.74 seconds
itlab1006@itlab1006-desktop:~$ sudo nmap -sF tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:23 IST
Nmap scan report for tsec.edu (103.211.218.165)
Host is up (0.029s latency).
rDNS record for 103.211.218.165: 103-211-218-165.webhostbox.net
All 1000 scanned ports on tsec.edu (103.211.218.165) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 30.90 seconds
itlab1006@itlab1006-desktop:~$ sudo nmap -sX google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:24 IST
Nmap scan report for google.com (142.250.71.110)
Host is up (0.0032s latency).
Other addresses for google.com (not scanned): 2404:6800:4009:806::200e
rDNS record for 142.250.71.110: pnbomb-ad-in-f14.1e100.net
All 1000 scanned ports on google.com (142.250.71.110) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 11.38 seconds
```

5)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.



6)IP protocol Scan-IP protocol scan works in a similar fashion to UDP scan. Instead of iterating through the port number field of a UDP packet, it sends IP packet headers and iterates through the eight-bit IP protocol field. The headers are usually empty, containing no data and not even the proper header for the claimed protocol. The exceptions are TCP, UDP, ICMP, SCTP, and IGMP.



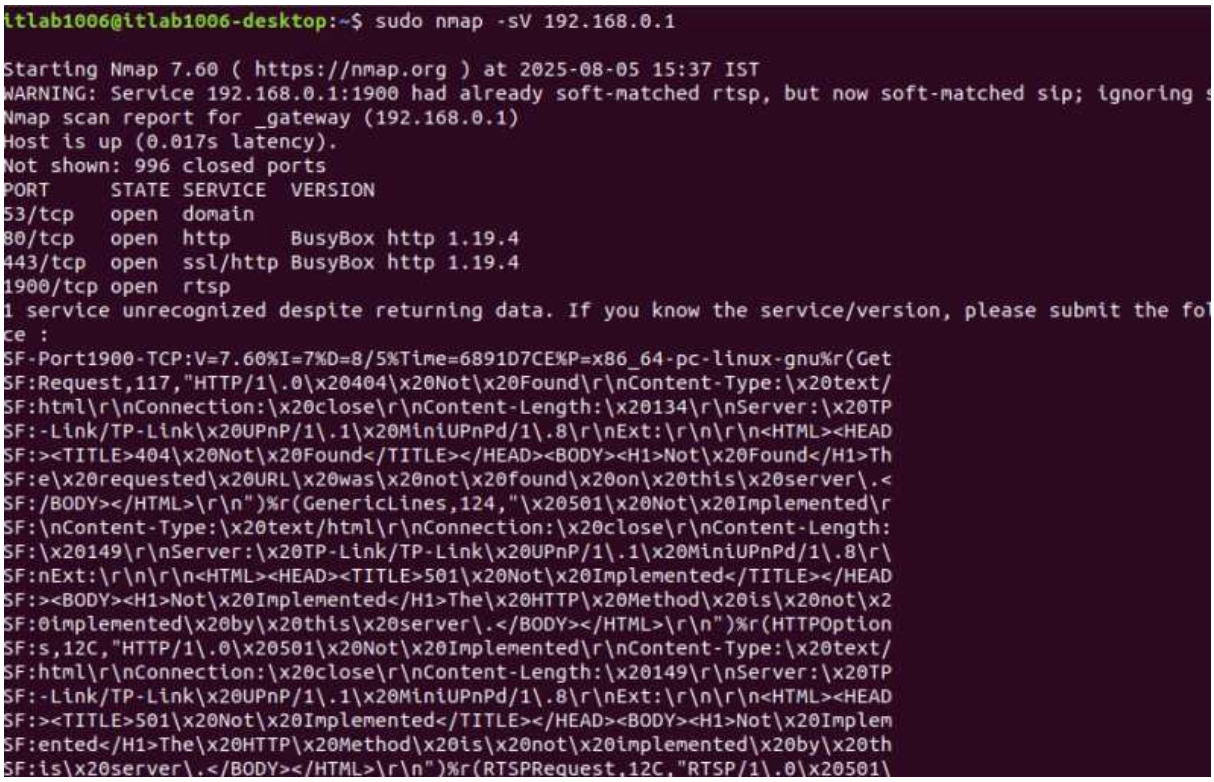
7)Port Ranges Scan- This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are ranges separated by a hyphen (e.g. 1- 1023). The beginning and/or end values of a range may be omitted, causing Nmap to use 1 and 65535, respectively. So you can specify -p- to scan ports from 1 through 65535. Scanning port zero is allowed if you specify it explicitly. For IP protocol scanning (-sO), this option specifies the protocol numbers you wish to scan for (0–255).

```
itlab1006@itlab1006-desktop:~$ sudo nmap -p 1000-2000 tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:31 IST
Nmap scan report for tsec.edu (103.211.218.165)
Host is up (0.029s latency).
rDNS record for 103.211.218.165: 103-211-218-165.webhostbox.net
All 1001 scanned ports on tsec.edu (103.211.218.165) are filtered

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

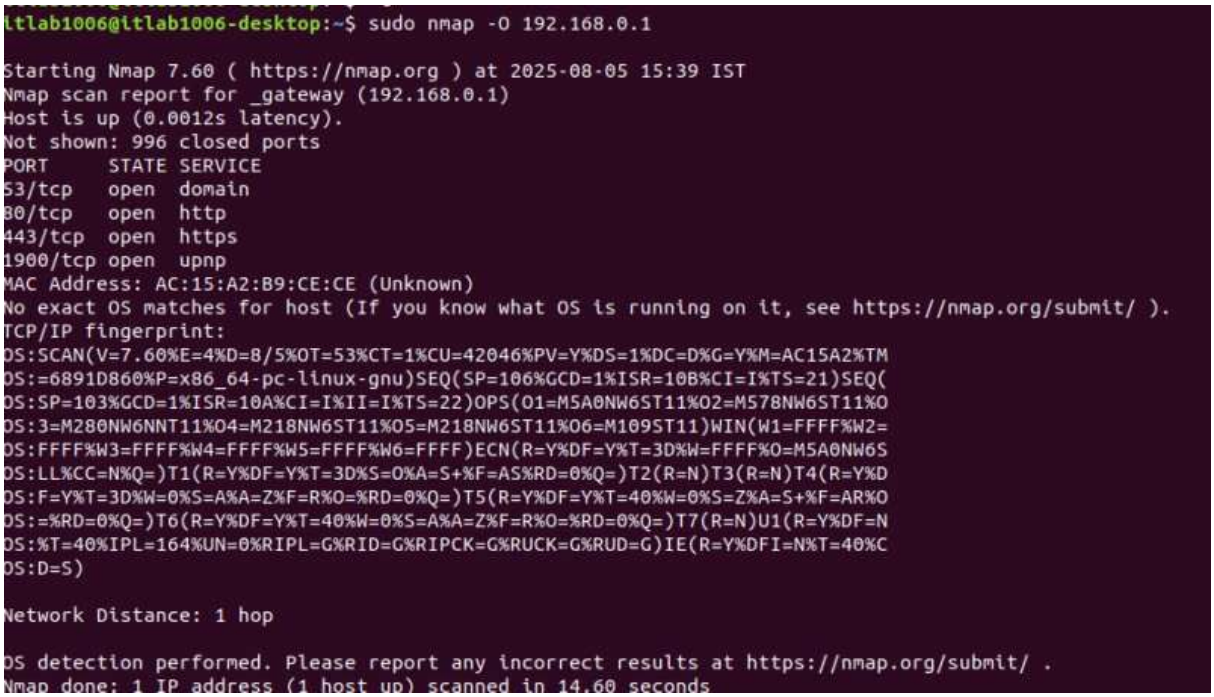
8)Fast scan-Specifies that you wish to scan fewer ports than the default. Normally Nmap scans the most common 1,000 ports for each scanned protocol. With -F, this is reduced to 100.

```
itlab1006@itlab1006-desktop:~$ sudo nmap -F tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2025-08-05 15:33 IST
Nmap scan report for tsec.edu (103.211.218.165)
Host is up (0.032s latency).
rDNS record for 103.211.218.165: 103-211-218-165.webhostbox.net
Not shown: 78 filtered ports
PORT      STATE  SERVICE
21/tcp    closed ftp
22/tcp    open   ssh
25/tcp    open   smtp
26/tcp    closed rsftp
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
8080/tcp  closed http-proxy
8443/tcp  closed https-alt
49152/tcp closed unknown
49153/tcp closed unknown
49154/tcp closed unknown
49155/tcp closed unknown
49156/tcp closed unknown
49157/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
```

9)Version Detection-Enables version detection, as discussed above. Alternatively, you can use -A, which enables version detection among other things



10)OS Scan- 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.



CONCLUSION:

Hence, we have installed NMAP and used it for various operations llike scanning, etc.

LO Mapping: LO-4

ASSIGNMENT 9

AIM:

Simulate DOS attack using Hping3.

THEORY:

Hping3 is a command-line network tool used for packet crafting and testing. It supports

TCP, UDP, ICMP, and RAW-IP protocols, and is often used for security auditing, firewall

testing, and network diagnostics.

A Denial of Service (DoS) attack aims to make a system or network resource unavailable

to its intended users by overwhelming it with traffic or exploiting vulnerabilities.

Hping3 in DoS Attacks

1. TCP SYN Flood

• Hping3 can send a massive number of SYN packets to a target.

• The target system allocates resources for each half-open connection.

• Eventually, it exhausts memory or CPU, causing service disruption.

2. ICMP Flood

• Hping3 sends continuous ICMP echo requests (like ping).

• The target spends resources replying, leading to performance degradation.

3. UDP Flood

• Hping3 sends large volumes of UDP packets to random ports.

• The target responds with ICMP “port unreachable” messages, consuming

bandwidth and CPU.

4. Spoofed Source IPs

• Attackers can use Hping3 to spoof IP addresses, making it harder to trace the

origin.

• This also amplifies the attack by confusing the target's response logic.

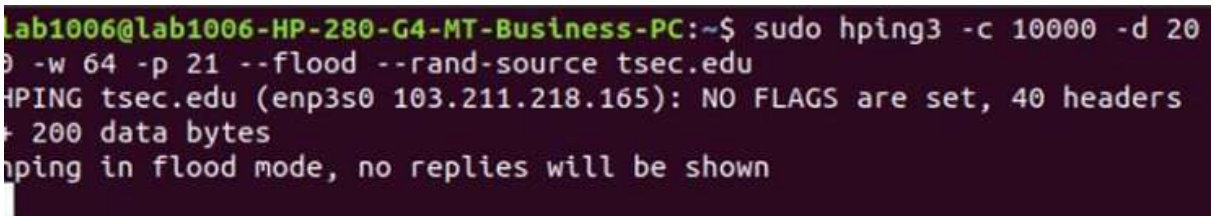
Mitigation Techniques

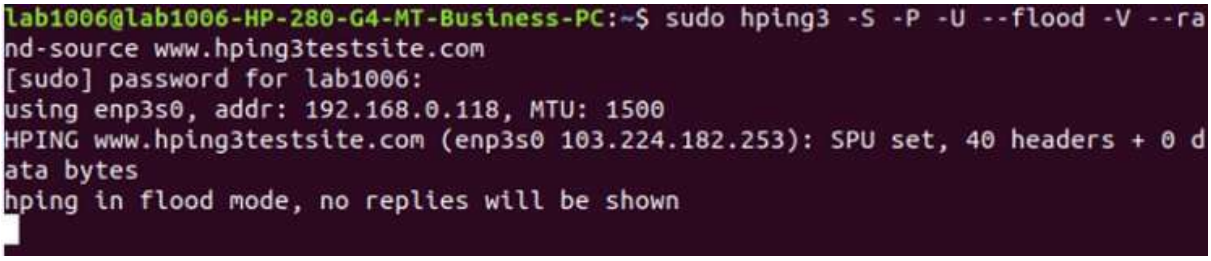
• Rate limiting and firewall rules to block suspicious traffic.

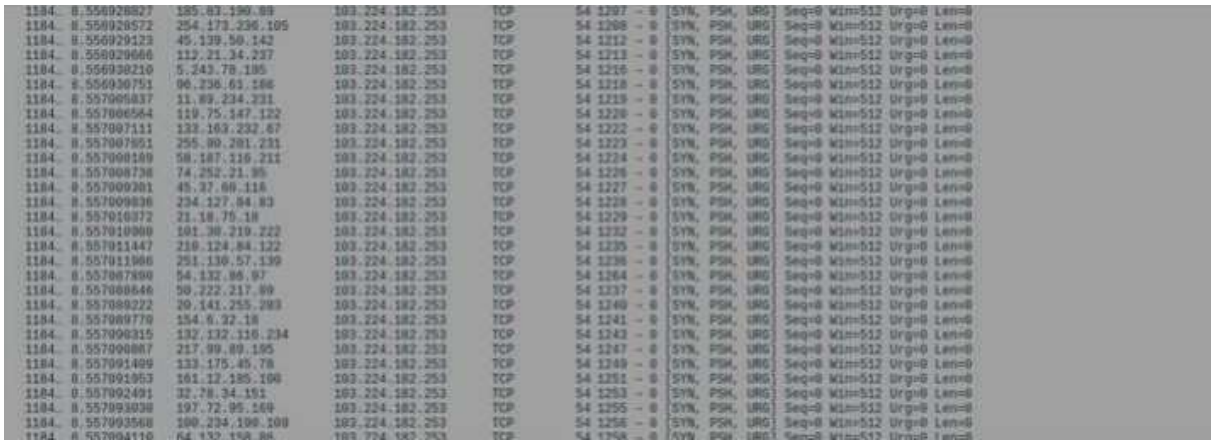
• Intrusion Detection Systems (IDS) to monitor packet patterns.

• TCP SYN cookies to prevent resource exhaustion.

SCREENSHOTS:







CONCLUSION:

Hence, we have understood and simulated a DOS attack using Hping3.

LO Mapping: LO-5

ASSIGNMENT 10

AIM:

To study and configure Firewalls using IP tables

THEORY:

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.

SCREENSHOTS:

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
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
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -p tcp --dport ssh -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
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
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -p tcp --dport telnet -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -p tcp --dport ftp -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -p tcp --dport http -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
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
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -p tcp --dport https -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -j DROP
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# ping 192.168.0.178
PING 192.168.0.178 (192.168.0.178) 56(84) bytes of data.
^C
--- 192.168.0.178 ping statistics ---
55 packets transmitted, 0 received, 100% packet loss, time 55290ms
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -I INPUT 6  -p icmp -j ACCEPT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# ping 192.168.0.119
PING 192.168.0.119 (192.168.0.119) 56(84) bytes of data.
64 bytes from 192.168.0.119: icmp_seq=1 ttl=64 time=0.645 ms
64 bytes from 192.168.0.119: icmp_seq=2 ttl=64 time=0.532 ms
64 bytes from 192.168.0.119: icmp_seq=3 ttl=64 time=0.539 ms
64 bytes from 192.168.0.119: icmp_seq=4 ttl=64 time=0.540 ms
64 bytes from 192.168.0.119: icmp_seq=5 ttl=64 time=0.250 ms
64 bytes from 192.168.0.119: icmp_seq=6 ttl=64 time=0.532 ms
64 bytes from 192.168.0.119: icmp_seq=7 ttl=64 time=0.500 ms
64 bytes from 192.168.0.119: icmp_seq=8 ttl=64 time=0.440 ms
64 bytes from 192.168.0.119: icmp_seq=9 ttl=64 time=0.532 ms
64 bytes from 192.168.0.119: icmp_seq=10 ttl=64 time=0.262 ms
64 bytes from 192.168.0.119: icmp_seq=11 ttl=64 time=0.247 ms
64 bytes from 192.168.0.119: icmp_seq=12 ttl=64 time=0.256 ms
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -s 192.168.0.109 -p icmp -j REJECT
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
REJECT     icmp --  192.168.0.109        anywhere             reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -A INPUT -s 192.168.0.109 -p icmp -j DROP
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
REJECT     icmp --  192.168.0.109        anywhere             reject-with icmp-port-unreachable
DROP       icmp --  192.168.0.109        anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

CONCLUSION:

Hence, we have understood and implemented IPTABLES, a robust and flexible firewall utility built into Linux systems, enabling precise control over network traffic.

LO Mapping: LO-6

ASSIGNMENT 11

AIM:

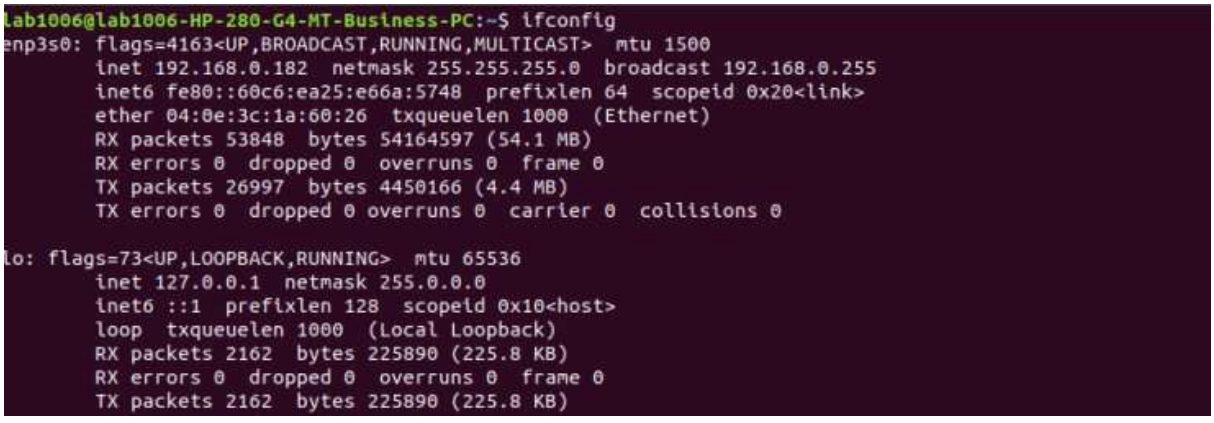
Installing snort, setting in Instruction Detection Mode and writing rules for intrusion detection.

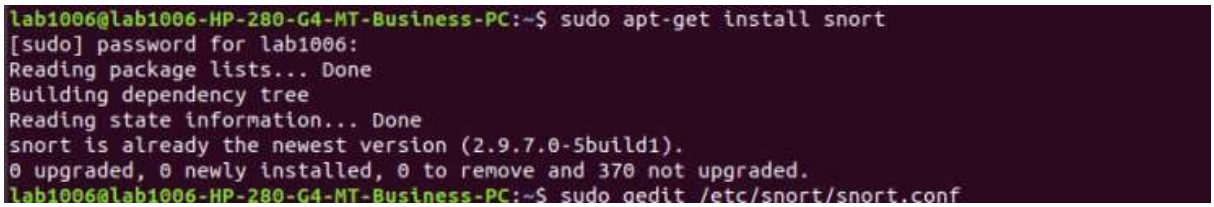
THEORY:

Hashdeep is a command-line tool in Linux used to compute and verify cryptographic hashes like MD5, SHA1, and SHA256 across files and directories. It supports recursive scanning, making it ideal for checking large datasets or backups for integrity. You can generate hashsets, compare them later, and detect any tampering or corruption with high precision.

It's especially useful in forensic analysis, system audits, and secure file verification workflows. Hashdeep's audit mode allows you to compare current files against a known hashset, flagging mismatches or missing files. A typical command looks like: hashdeep -r -c md5,sha1,sha256 -l -o t folder/ , which recursively hashes files and outputs a timestamped log.

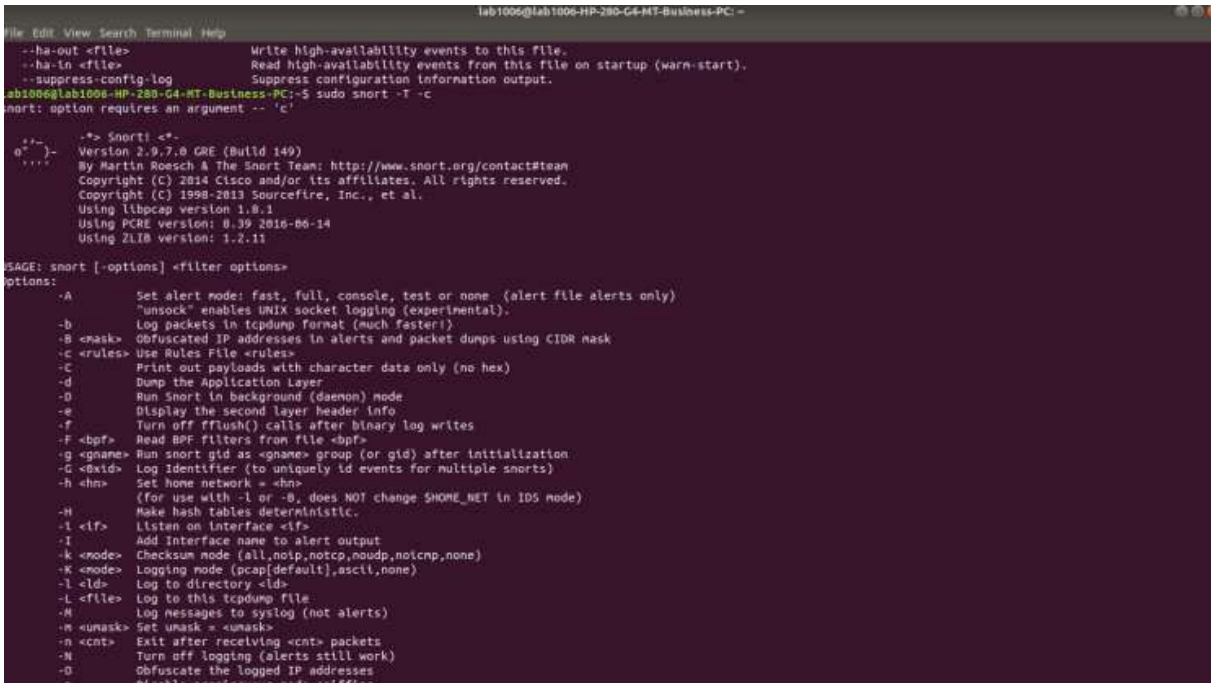
SCREENSHOTS:

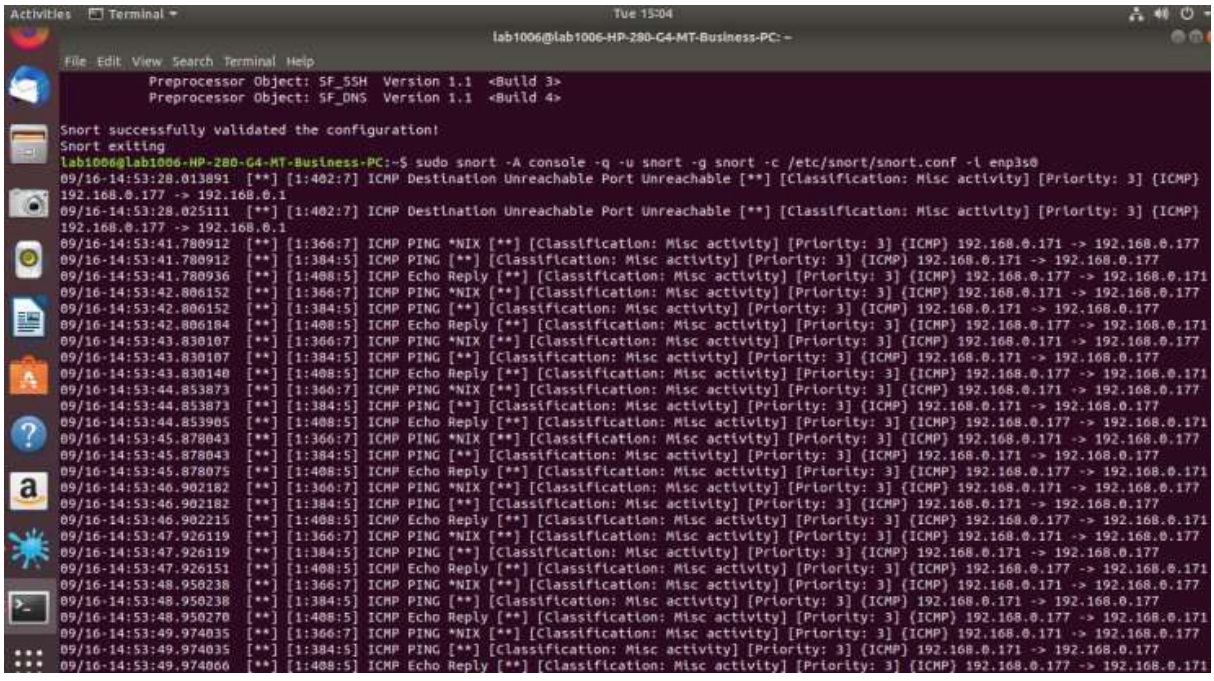




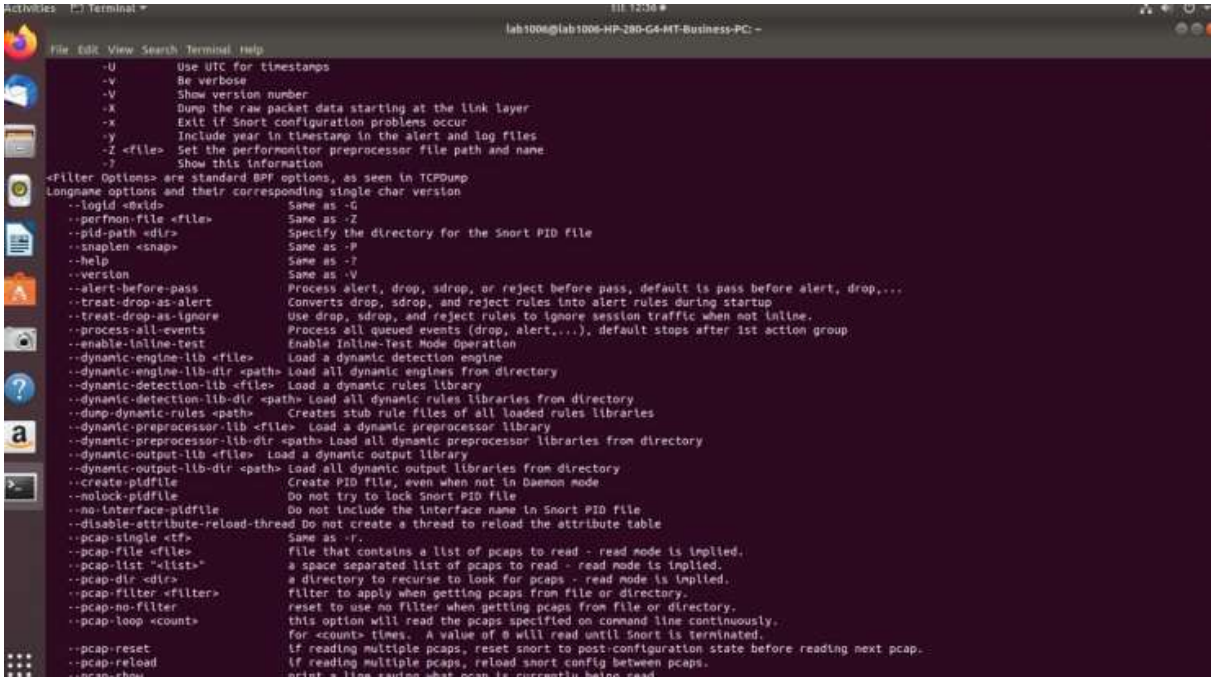
```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -T -c
snort: option requires an argument -- 'c'

          -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.8.1
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

USAGE: snort [-options] <filter options>
Options:
        -A          Set alert mode: fast, full, console, test or none  (alert file alerts only)
                    "unsock" enables UNIX socket logging (experimental).
        -b          Log packets in tcpdump format (much faster!)
        -B <mask>   Obfuscated IP addresses in alerts and packet dumps using CIDR mask
        -c <rules>  Use Rules File <rules>
        -C          Print out payloads with character data only (no hex)
        -d          Dump the Application Layer
        -D          Run Snort in background (daemon) mode
        -e          Display the second layer header info
        -f          Turn off fflush() calls after binary log writes
        -F <bpf>    Read BPF filters from file <bpf>
        -g <gname>  Run snort gid as <gname> group (or gid) after initialization
        -G <0xid>   Log Identifier (to uniquely id events for multiple snorts)
        -h <hn>     Set home network = <hn>
                    (for use with -l or -B, does NOT change $HOME_NET in IDS mode)
        -H          Make hash tables deterministic.
        -i <if>     Listen on interface <if>
        -I          Add Interface name to alert output
        -k <mode>   Checksum mode (all,noip,notcp,noudp,noicmp,none)
```

```
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
  --logid <0xid>                  Same as -G
  --perfmon-file <file>           Same as -Z
  --pid-path <dir>                Specify the directory for the Snort PID file
  --snaplen <snap>                Same as -P
  --help                          Same as -?
  --version                       Same as -V
  --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
  --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
  --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
  --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
  --enable-inline-test            Enable Inline-Test Mode Operation
  --dynamic-engine-lib <file>     Load a dynamic detection engine
  --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
  --dynamic-detection-lib <file>  Load a dynamic rules library
  --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
  --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
  --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
  --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
  --dynamic-output-lib <file>  Load a dynamic output library
  --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
  --create-pidfile                Create PID file, even when not in Daemon mode
  --nolock-pidfile                Do not try to lock Snort PID file
  --no-interface-pidfile          Do not include the interface name in Snort PID file
  --disable-attribute-reload-thread Do not create a thread to reload the attribute table
  --pcap-single <tf>              Same as -r.
  --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
  --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
  --pcap-dir <dir>                a directory to recurse to look for pcaps - read mode is implied.
  --pcap-filter <filter>          filter to apply when getting pcaps from file or directory.
  --pcap-no-filter                reset to use no filter when getting pcaps from file or directory.
  --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                  for <count> times.  A value of 0 will read until Snort is terminated.
  --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
  --pcap-reload                   if reading multiple pcaps, reload snort config between pcaps.
  --pcap-show                     print a line saying what pcap is currently being read.
  --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
```


lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
--ha-out <file> Write high-availability events to this file.
--ha-in <file> Read high-availability events from this file on startup (warn-start).
--suppress-config-log Suppress configuration information output.
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -T -c
snort: option requires an argument -- 'c'
-*> Snort! <*-
Version 2.9.7.0 GRE (Build 149)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.8.1
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11
USAGE: snort [-options] <filter options>
Options:
-A Set alert mode: fast, full, console, test or none (alert file alerts only)
"unsock" enables UNIX socket logging (experimental).
-b Log packets in tcpdump format (much faster!)
-B <mask> Obfuscated IP addresses in alerts and packet dumps using CIDR mask
-c <rules> Use Rules File <rules>
-C Print out payloads with character data only (no hex)
-d Dump the Application Layer
-D Run Snort in background (daemon) mode
-e Display the second layer header info
-f Turn off fflush() calls after binary log writes
-F <bpf> Read BPF filters from file <bpf>
-g <gname> Run snort gid as <gname> group (or gid) after initialization
-G <0xid> Log Identifier (to uniquely id events for multiple snorts)
-h <hn> Set home network = <hn>
(for use with -l or -B, does NOT change $HOME_NET in IDS mode)
-H Make hash tables deterministic.
-i <if> Listen on interface <if>
-I Add Interface name to alert output
-k <mode> Checksum mode (all,noip,notcp,noudp,noicmp,none)
-K <mode> Logging mode (pcap[default],ascii,none)
-l <ld> Log to directory <ld>
-L <file> Log to this tcpdump file
-M Log messages to syslog (not alerts)
-m <umask> Set umask = <umask>
-n <cnt> Exit after receiving <cnt> packets
-N Turn off logging (alerts still work)
-O Obfuscate the logged IP addresses


Activities Terminal Tue 15:04
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Snort successfully validated the configuration!
Snort exiting
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp3s0
09/16-14:53:28.013891 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.1
09/16-14:53:28.025111 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.1
09/16-14:53:41.780912 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:41.780912 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:41.780936 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:42.806152 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:42.806152 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:42.806184 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:43.830107 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:43.830107 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:43.830140 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:44.853873 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:44.853873 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:44.853905 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:45.878043 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:45.878043 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:45.878075 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:46.902182 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:46.902182 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:46.902215 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:47.926119 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:47.926119 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:47.926151 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:48.950238 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:48.950238 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:48.950270 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171
09/16-14:53:49.974035 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:49.974035 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.171 -> 192.168.0.177
09/16-14:53:49.974066 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.177 -> 192.168.0.171

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -T -c /etc/snort/snort.conf -i enp3s0
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 71
44:7145 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 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 9
090:9091 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_dynamicengine/libsf_engine.so... done
Loading all dynamic detection libs from /usr/lib/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort_dynamicrules
```

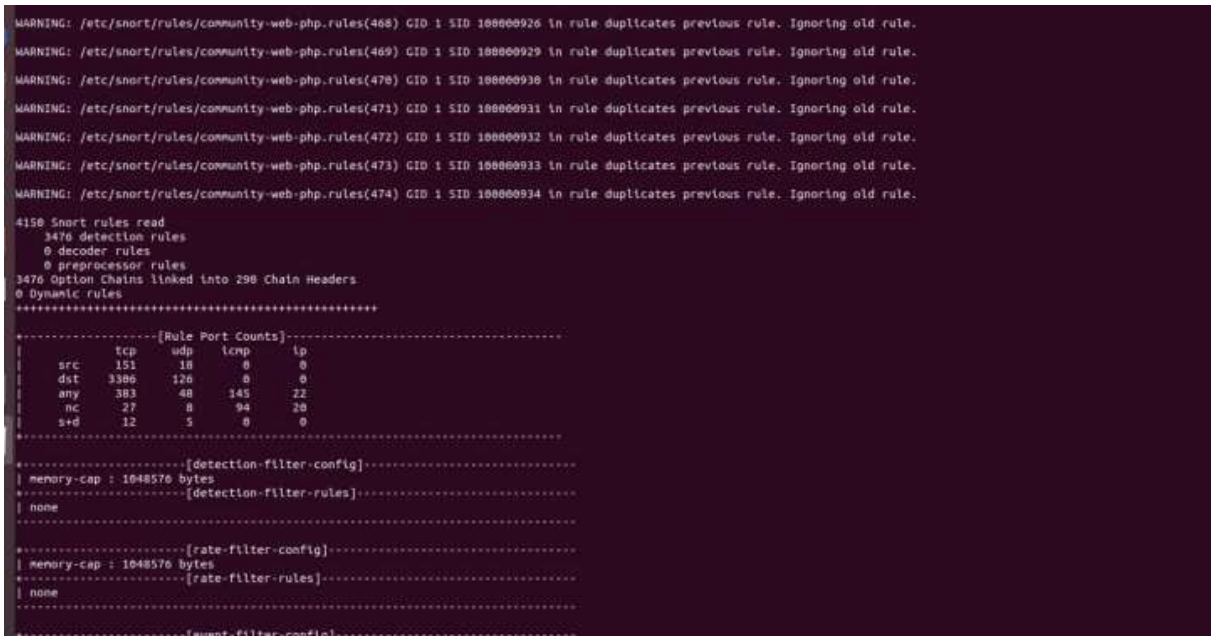


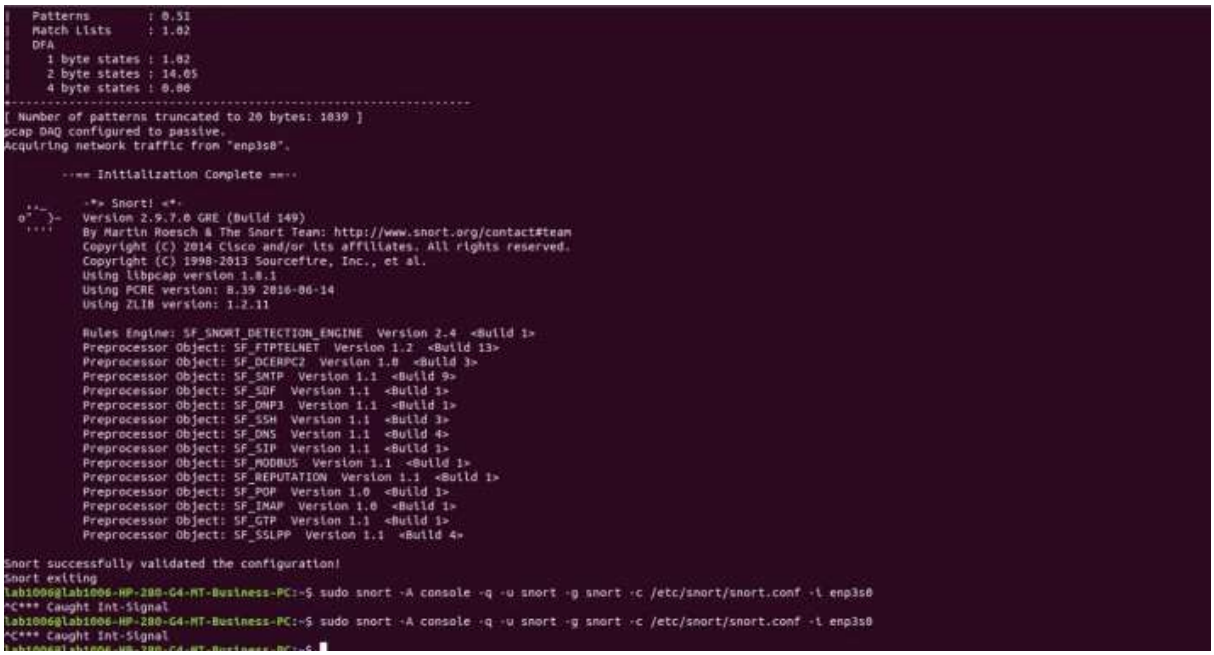
```
File  Edit  View  Search  Terminal  Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -T -C /etc/snort/snort.conf -i enp3s0
ERROR: Test mode must be run with a snort configuration file.  Use the '-c' option on the command line to specify a configuration file.
Fatal Error, Quitting..
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo snort -T -c /etc/snort/snort.conf -i enp3s0
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 80
0 8008 8014 8020 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 6988 7000:7001 7144:7145 7510
7777 7779 8000 8008 8014 8020 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 55
55 ]
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_dynamicengine/libsf_engine.so... done
Loading all dynamic detection libs from /usr/lib/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/...
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_ssl_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_gtp_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_imap_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_pop_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_reputation_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_modbus_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_sip_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dns_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_ssh_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dnp3_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_sdf_preproc.so... done
```

```
    Ports:
        502
DNP3 config:
    Memcap: 262144
    Check Link-Layer CRCs: ENABLED
    Ports:
        20000

++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
WARNING: /etc/snort/rules/chat.rules(33) threshold (in rule) is deprecated; use detection_filter instead.

WARNING: /etc/snort/rules/community-sql-injection.rules(6) GID 1 SID 100000106 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(7) GID 1 SID 100000107 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(8) GID 1 SID 100000108 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(9) GID 1 SID 100000109 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(11) GID 1 SID 100000192 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(12) GID 1 SID 100000193 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(13) GID 1 SID 100000194 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(14) GID 1 SID 100000690 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-sql-injection.rules(15) GID 1 SID 100000691 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-web-client.rules(6) GID 1 SID 100000118 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-web-client.rules(7) GID 1 SID 100000119 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-web-client.rules(9) GID 1 SID 100000228 in rule duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-web-client.rules(14) GID 1 SID 100000284 in rule duplicates previous rule. Ignoring old rule.
```

```
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_imap_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_pop_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_reputation_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_modbus_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_sip_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dns_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_ssh_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dnp3_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_sdf_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_smtp_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dce2_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_ftptelnet_preproc.so... done
Finished Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/
og directory = /var/log/snort
ARNING: ip4 normalizations disabled because not inline.
ARNING: tcp normalizations disabled because not inline.
ARNING: icmp4 normalizations disabled because not inline.
ARNING: ip6 normalizations disabled because not inline.
ARNING: icmp6 normalizations disabled because not inline.
rag3 global config:
   Max frags: 65536
   Fragment memory cap: 4194304 bytes
rag3 engine config:
   Bound Address: default
   Target-based policy: WINDOWS
   Fragment timeout: 180 seconds
   Fragment min_ttl:   1
   Fragment Anomalies: Alert
   Overlap Limit:     10
   Min fragment Length:     100
     Max Expected Streams: 768
tream global config:
   Track TCP sessions: ACTIVE
   Max TCP sessions: 262144
   TCP cache pruning timeout: 30 seconds
   TCP cache nominal timeout: 3600 seconds
   Memcap (for reassembly packet storage): 8388608
   Track UDP sessions: ACTIVE
   Max UDP sessions: 131072
   UDP cache pruning timeout: 30 seconds
   UDP cache nominal timeout: 180 seconds
   Track ICMP sessions: INACTIVE
   Track IP sessions: INACTIVE
   Log info if session memory consumption exceeds 1048576
```

```
   alert_fragments: INACTIVE
   alert_large_fragments: INACTIVE
   alert_incomplete: INACTIVE
   alert_multiple_requests: INACTIVE
TPTelnet Config:
   GLOBAL CONFIG
     Inspection Type: stateful
     Check for Encrypted Traffic: YES alert: NO
     Continue to check encrypted data: YES
   TELNET CONFIG:
     Ports: 23
     Are You There Threshold: 20
     Normalize: YES
     Detect Anomalies: YES
   FTP CONFIG:
     FTP Server: default
       Ports (PAF): 21 2100 3535
       Check for Telnet Cmds: YES alert: YES
       Ignore Telnet Cmd Operations: YES alert: YES
       Ignore open data channels: NO
     FTP Client: default
       Check for Bounce Attacks: YES alert: YES
       Check for Telnet Cmds: YES alert: YES
       Ignore Telnet Cmd Operations: YES alert: YES
       Max Response Length: 256
MTP Config:
   Ports: 25 465 587 691
   Inspection Type: Stateful
   Normalize: ATRN AUTH BDAT DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND STARTTLS SOML TICK TIME TU
N TURNME VERB VRFY X-EXPS XADR XAUTH XCIR XEXCH50 XGEN XLICENSE X-LINK2STATE XQUE XSTA XTRN XUSR CHUNKING X-ADAT X-DRCP X-ERCP X-EXCH50
   Ignore Data: No
   Ignore TLS Data: No
   Ignore SMTP Alerts: No
   Max Command Line Length: 512
   Max Specific Command Line Length:
     ATRN:255 AUTH:246 BDAT:255 DATA:246 DEBUG:255
     EHLO:500 EMAL:255 ESAM:255 ESND:255 ESOM:255
     ETRN:246 EVFY:255 EXPN:255 HELO:500 HELP:500
     IDENT:255 MAIL:260 NOOP:255 ONEX:246 QUEU:246
     QUIT:246 RCPT:300 RSET:246 SAML:246 SEND:246
     SIZE:255 STARTTLS:246 SOML:246 TICK:246 TIME:246
     TURN:246 TURNME:246 VERB:246 VRFY:255 X-EXPS:246
     XADR:246 XAUTH:246 XCIR:246 XEXCH50:246 XGEN:246
     XLICENSE:246 X-LINK2STATE:246 XQUE:246 XSTA:246 XTRN:246
     XUSR:246
```





CONCLUSION:

Hence, we have understood and implemented snort, and used it in Intrusion Detection Mode.

LO Mapping: LO-6

ASSIGNMENT 12

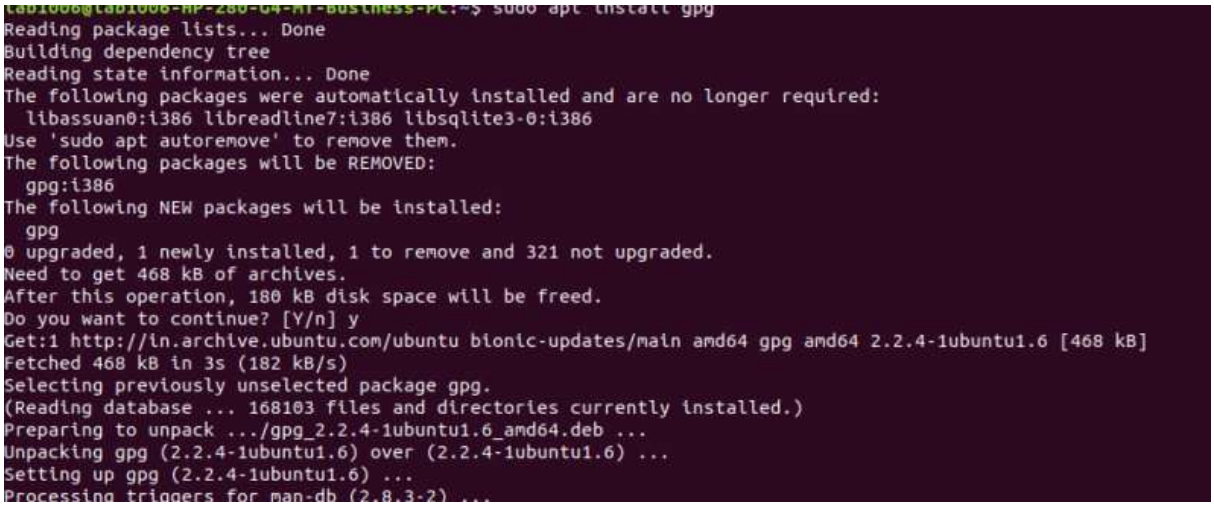
AIM:

Explore the GPG tool of Linux and ensure email security.

THEORY:

GPG (GNU Privacy Guard) is a Linux tool used for encrypting, decrypting, and signing files using public-key cryptography. You can generate a key pair with gpg --fullgenerate-key, encrypt files using gpg -e -r recipient@example.com file.txt, and decrypt them with gpg -d file.txt.gpg. It's widely used for secure communication, verifying software, and protecting sensitive data.

SCREENSHOTS:



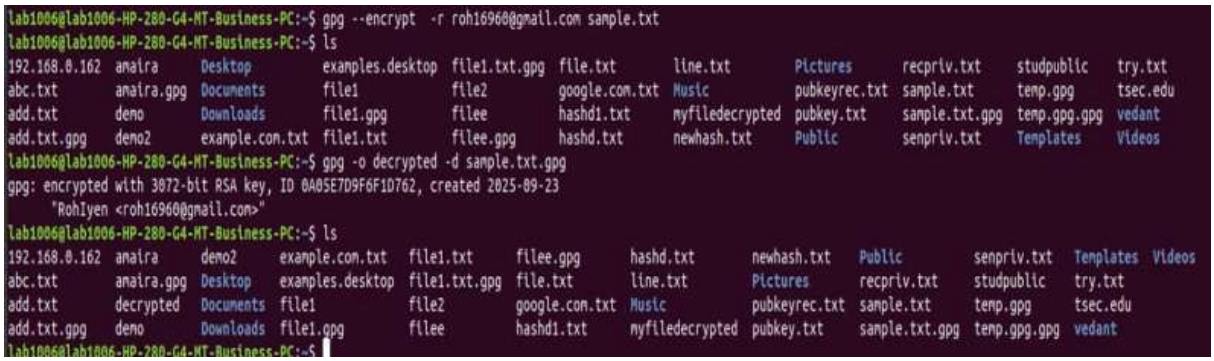
```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --gen-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

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: Rohit
Email address: rohitiyengar29@gmail.com
You selected this USER-ID:
    "Rohit <rohitiyengar29@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 525FA421A585C554 marked as ultimately trusted
gpg: revocation certificate stored as '/home/lab1006/.gnupg/openpgp-revocs.d/92046A32B98C18C50E9A6820525FA421A585C554.rev'
public and secret key created and signed.

pub   rsa3072 2025-09-23 [SC] [expires: 2027-09-23]
      92046A32B98C18C50E9A6820525FA421A585C554
uid                      Rohit <rohitiyengar29@gmail.com>
sub   rsa3072 2025-09-23 [E] [expires: 2027-09-23]
```

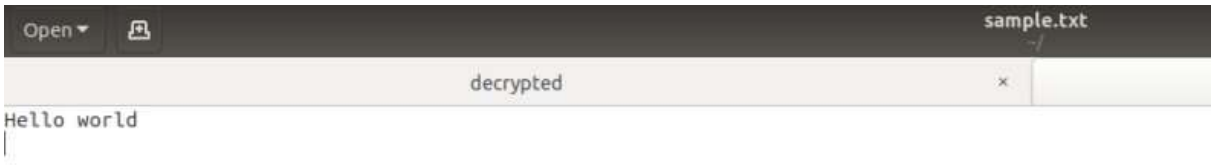

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --gen-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

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: Rohit
Email address: rohitiyengar29@gmail.com
You selected this USER-ID:
    "Rohit <rohitiyengar29@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 525FA421A585C554 marked as ultimately trusted
gpg: revocation certificate stored as '/home/lab1006/.gnupg/openpgp-revocs.d/92046A32B98C18C50E9A6820525FA421A585C554.rev'
public and secret key created and signed.

pub   rsa3072 2025-09-23 [SC] [expires: 2027-09-23]
      92046A32B98C18C50E9A6820525FA421A585C554
uid                      Rohit <rohitiyengar29@gmail.com>
sub   rsa3072 2025-09-23 [E] [expires: 2027-09-23]
```


```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --output pubkey.txt --armor --export rohitiyengar29@gmail.com
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --output pubkeyrec.txt --armor --export roh16960@gmail.com
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ ls
192.168.0.162  anaira      Desktop          examples.desktop  file1.txt.gpg  file.txt       line.txt        Pictures       studpublic    try.txt
abc.txt        anaira.gpg  Documents        file1             file2          google.com.txt Music           pubkeyrec.txt  temp.gpg      tsec.edu
add.txt        demo        Downloads        file1.gpg         filee          hashd1.txt     myfiledecrypted pubkey.txt     temp.gpg.gpg  vedant
add.txt.gpg    demo2       example.com.txt  file1.txt         filee.gpg      hashd.txt      newhash.txt     Public         Templates     Videos
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --export-secret-key -a Rohit>senpriv.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --export-secret-key -a RohIyen>recpriv.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ ls
192.168.0.162  anaira      Desktop          examples.desktop  file1.txt.gpg  file.txt       line.txt        Pictures       recpriv.txt   temp.gpg.gpg  vedant
abc.txt        anaira.gpg  Documents        file1             file2          google.com.txt Music           pubkeyrec.txt  senpriv.txt   Templates     Videos
add.txt        demo        Downloads        file1.gpg         filee          hashd1.txt     myfiledecrypted pubkey.txt     studpublic    try.txt
add.txt.gpg    demo2       example.com.txt  file1.txt         filee.gpg      hashd.txt      newhash.txt     Public         temp.gpg      tsec.edu
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --fingerprint rohitiyengar29@gmail.com
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   8  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 8u
gpg: next trustdb check due at 2026-10-04
pub   rsa3072 2025-09-23 [SC] [expires: 2027-09-23]
      9204 6A32 B98C 18C5 0E9A  6820 525F A421 A585 C554
uid           [ultimate] Rohit <rohitiyengar29@gmail.com>
sub   rsa3072 2025-09-23 [E] [expires: 2027-09-23]

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --fingerprint roh16960@gmail.com
pub   rsa3072 2025-09-23 [SC] [expires: 2027-09-23]
      6611 C6AF B5E3 930D 23FE  136B 387D 0264 0202 7AEE
uid           [ultimate] RohIyen <roh16960@gmail.com>
sub   rsa3072 2025-09-23 [E] [expires: 2027-09-23]
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --import pubkey.txt
gpg: key 525FA421A585C554: "Rohit <rohitiyengar29@gmail.com>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --import pubkeyrec.txt
gpg: key 387D026402027AEE: "RohIyen <roh16960@gmail.com>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --list keys
gpg: option "--list" is ambiguous
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ gpg --list-keys
/home/lab1006/.gnupg/pubring.kbx
--------------------------------
pub   rsa3072 2024-10-04 [SC] [expires: 2026-10-04]
      660F68BEA72E9C91B9EB68E3FFE19C6CEFF88DB3
uid           [ultimate] Amaira <vaziraniamaira@gmail.com>
sub   rsa3072 2024-10-04 [E] [expires: 2026-10-04]

pub   rsa3072 2024-10-08 [SC] [expires: 2026-10-08]
      FF5675EC7CD9626DA8587B33926F8B37241E8C03
uid           [ultimate] madam <madam123@gmsil.com>
sub   rsa3072 2024-10-08 [E] [expires: 2026-10-08]

pub   rsa3072 2024-10-08 [SC] [expires: 2026-10-08]
      420F6951FE5A230A2C3E2230133EA06C52A728AF
uid           [ultimate] student <student@gmail.com>
sub   rsa3072 2024-10-08 [E] [expires: 2026-10-08]

pub   rsa3072 2024-10-08 [SC] [expires: 2026-10-08]
      CCE2C689C62E6A8E2DE70969D0DA2B8371087D47
uid           [ultimate] lolol <lol@gmail.com>
sub   rsa3072 2024-10-08 [E] [expires: 2026-10-08]

pub   rsa3072 2024-10-10 [SC] [expires: 2026-10-10]
      D2BCFA27EE9245418B84BE973A3E76E3C5E4511C
uid           [ultimate] sweetu <sweetuuunarsinghani@gmail.com>
sub   rsa3072 2024-10-10 [E] [expires: 2026-10-10]

pub   rsa3072 2024-10-10 [SC] [expires: 2026-10-10]
      55F43739D91BB55C18CB3B278C36612872005B90
uid           [ultimate] saloni <abcd@gmail.com>
sub   rsa3072 2024-10-10 [E] [expires: 2026-10-10]

pub   rsa3072 2025-09-23 [SC] [expires: 2027-09-23]
```





CONCLUSION:

Hence, we have understood and implemented email security using GPG in Ubuntu.

LO Mapping: LO-6

AYUSH SHARMA
T22-117

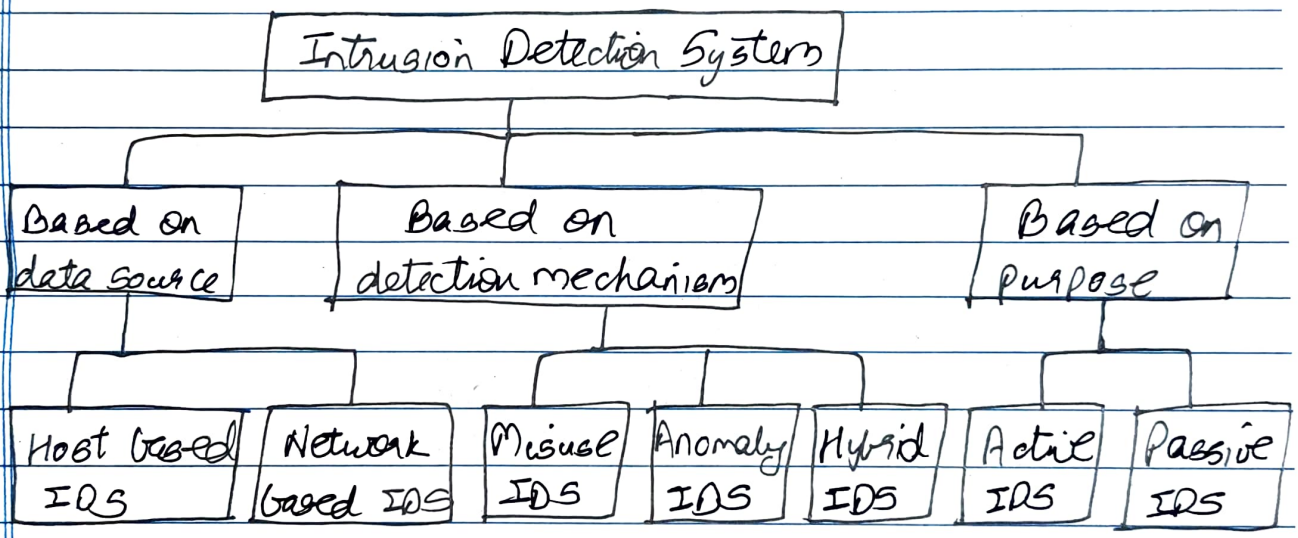
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

SL WRITTEN ASSIGNMENT-2

Q1. What is IDS? Explain its types, advantages/disadvantages and applications.

Ans.

→ An intrusion detection system (IDS) is a security mechanism that monitors & analyzes network or system activities to detect malicious activities, policy violations or unauthorized access attempts



- Types of IDS:

→ Based on data source

① Network based IDS (NIDS) :- Monitors entire network traffic

② Host based IDS (HIDS) :- Monitors activities of a specific host

→ Based on detection method:

① ~~Signature~~ Anomaly based IDS :- Detects deviations from normal behaviours.

② ~~Anomaly~~ Signature based IDS :- Detects attacks by matching patterns

③ Hybrid IDS :- Combination of both signature & anomaly methods.

→ Based on purpose

① Passive IPS :- Only generates alerts when a intrusion is detected.

② Active IPS :- Takes action

• Advantages

→ Provides early detection of suspicious activities
→ Helps prevent unauthorized access & data breaches
→ Monitors both internal & external threats
→ Provides forensic information for investigation
→ Increases overall system & network security

• Disadvantages:

→ High false positives
→ Cannot stop attacks directly
→ Requires constant updates for new attack signatures.
→ Resource intensive

• Applications :

→ Enterprise networks :- Monitoring large scale corporate systems
→ Government & defense :- Protecting sensitive information against attackers
→ Banking & finance :- Detecting fraud and unauthorized access.
→ Cloud & data centers :- Preventing attacks on hosted services
→ E-commerce :- Safeguarding transactions and customer data.

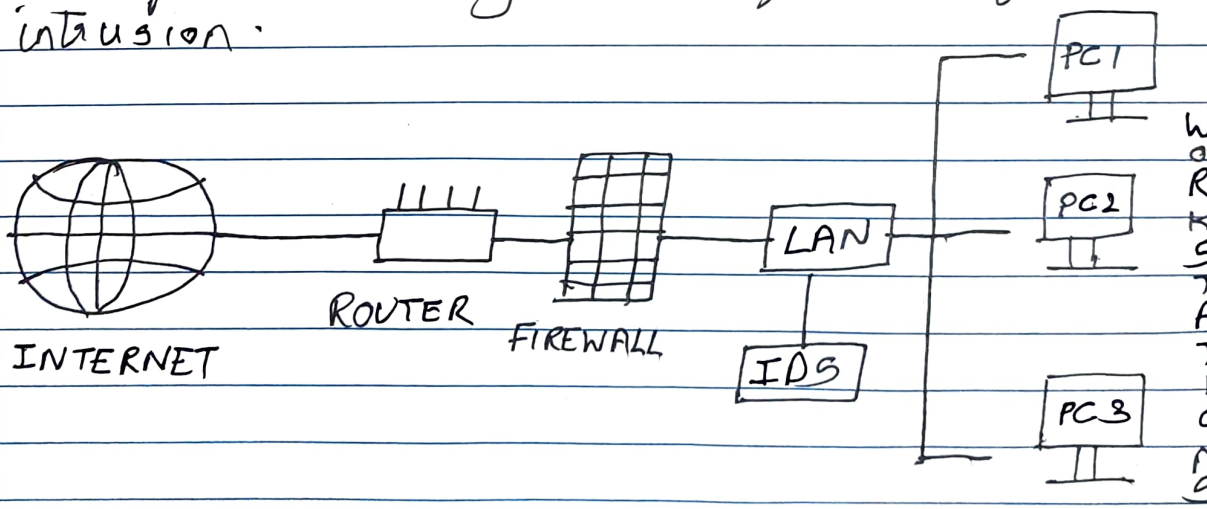
- Working of IDS

→ An IDS monitors the traffic on a computer network to detect any suspicious activities

→ It analyzes the data flowing through the network to look for patterns and signs of abnormal behaviours.

→ The IDS compares the network activity to a set of predefined rules and patterns to identify any activity that might indicate an attack or intrusion.

→ The system administrator can then investigate the alert and take action to prevent any damage or further intrusion.



THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Pradeep
06/04/25

★ Class Assignment ★

Q.1) Describe different types of DoS Attacks.

Ans:) 1) Denial of Service (DoS) Attack is a malicious attempt to make a computer, server or network resource unavailable to its intended users by overwhelming it with excessive traffic or exploiting its vulnerabilities.

1. Volume Based Attacks

Definition :-

These attacks flood the target with huge amounts of traffic, exhausting its bandwidth.

Examples :-

- UDP Flood :- Sends large numbers of User Datagram Protocol (UDP) packets to random ports.
- ICMP Flood :- (Ping Flood) :- Overloads the target with continuous ICMP Echo Request (ping) packets.

Goal

- Consume all available bandwidth.

THADOMAL SHA
TSEC
ENGINEERING COLLEGE

2 Protocol Attacks

Definition

These attacks exploit weaknesses in network protocols or server resources (like firewall and load balancers).

Examples

- SYN Flood :- Attacker sends repeated TCP "connection requests" but never completes the handshake, leaving the server stuck with half-open connections.
- Ping of Death :- Sends malformed or oversized packets that crash the target system
- Smurf Attack :- Uses spoofed ICMP requests sent to a network broadcast address, causing all devices to reply to the victim

Goal.

- Exhaust servers resources (CPU, memory, and connection tables)

3 Application layer Attacks

Definition :-

Target the application or software layer (layer 7 of the OSI model) by sending legitimate looking requests that overwhelm services.

- Examples

a) HTTP Flood :-

Sends a huge number of HTTP requests to a web server (like refreshing a webpage repeatedly).

b) Slowloris Attack :-

Sends partial HTTP requests and keeps connections open, exhausting the server's thread pool.

- Goal

Make web applications or specific services unavailable

4. Distributed Denial of Service (DDoS) Attacks

Definition

A large scale version of DoS, where multiple compromised devices attacks a single target simultaneously

Example

A DDoS attacks on a website by thousands of infected computers/ zombies.

Goal

Make it harder to block the attack since traffic comes from multiple sources.

TSEC
ENGINEERING COLLEGE

Q.2) Explain different elements used in NAC

Ans 2) Network Access Control (NAC) is a security solution that controls which devices and users can access a network. It ensures that only authorized, compliant, and secure devices are allowed in.

The main elements of NAC are:-

1. Policy Management

Definition

Policies define the rules for network access

Functions

- Set authentication rules (eg only employees with valid credentials can access).
- Defines compliance requirements (eg antivirus must be updated)
- Determines access levels (eg guest vs admin)

2. Authentication

Definition

Verifies the identity of users or devices before granting network access.

Methods

- Password / username
- Multifactor Authentication (MFA)

- Digital Certificates
- Biometric Authentication

Purpose :- Ensures only trusted users / devices can connect.

3 End Point Security

- Definition
  Checks the security status of a device before allowing it into the network
- Checks For
  a) Antivirus / Firewall status
  b) Latest Security Patches
  c) System configurations
- Purpose
  Prevents infected or non-compliant devices from spreading threats.

4 Access Enforcement

- Definition
  The mechanism that allows or denies network access based on the defined policy
- How it works
  a) Devices are either granted full access, restricted access (eg Internet only)
  b) Enforced using switches, routers, firewalls or wireless controllers.

5. Monitoring and Reporting

- Definition

  Continuous observation of devices and user activity on the network

- Functions

  a) logs all access attempts
  b) Detects unusual behaviors.
  c) Generates reports for compliance and audits.

Q.3) Explain SSL & TLS.

Ans 3) 1. SSL (Secure Sockets Layer)

Definition

SSL is a cryptographic protocol developed by Netscape in the mid 1990s to provide secure communication over the internet

Purpose

It encrypts data between a client and a server, preventing eavesdropping and tampering.

Features

- Provides encryption
- Provides authentication
- Provides data integrity.

Limitation

SSL is now considered outdated and insecure due to known vulnerabilities.

2. TLS (Transport layer Security)

Definition

TLS is the upgraded and more secure version of SSL, introduced by the IETF.

Purpose

like SSL, it secures communication over networks but with stronger encryption algorithms and improved security features

Features

- uses stronger cryptographic algorithms than SSL
- Prevents many known attacks
- Provides forward secrecy.
- Still widely used today in HTTPS, email security, VPNs etc

Q.4) Describe Transport Mode and Tunneling Mode in IP security (IPSec)

Ans 4) IPSec provides security at the network layer by protecting IP packets through authentication, integrity and confidentiality..

It works in two modes:- Transport Mode and Tunnel Mode

1 Transport Mode

In transport mode, only the payload of the IP packet is encrypted or authenticated. The

original IP header remains intact

Use Case

Typically used for end to end communication between two hosts.

Advantages

- less Overhead
- Efficient for direct communication between two devices.

Disadvantages

- Does not hide the source and destination IP addresses, so traffic analysis is still possible

2. Tunnel Mode

Definition

In tunnel mode, the entire original IP packet (header + payload) is encrypted and encapsulated inside a new IP packet with a fresh outer IP header

use Case

Typically used for network to network or host to network

- Advantages

Provides full protection, including original IP header

- Disadvantages

More overhead due to additional outer IP header