Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ SAIKARTHIK IYER

of IT Department, Semester V with

Roll No. 41 has completed a course of the necessary

experiments in the subject SL under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2024 - 2025

22/10/24

Teacher In- Charge

Head of the Department

Date 22/10/24

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1	Breaking Shift Cipher and mono alphabetic Substitution Cipher using frequency analysis method		18/7/24	
2	Cryptanalysis or decoding of polyalphabetic cipher: playfair and vigenere cipher.		25/2/24	
3	Block cipher mode of operation using AES.		1/8/24	
4	Implementation and analysis of RSA cryptosystem and digital signature scheme using RSA.		8/8/24	
5	To explore hashdeep tool in kali linux or generating, matching and auditing hash of files.		29/8/24	
6	Study the use of network reconnaissance tools like dig, whois traceroute, etc.		5/9/24	
7	Study packet sniffer tools.		12/9/24	22/10/24
8	Installation of NMAP and using it with different options.			
9	Simulate DOS attack using Hping3		19/9/24	
10	Study and configure firewall using IP tables		26/9/24	
11	Installing snort, setting in IDM and writing rules for IDS.		3/10/24	
12	Explore the GPG tool of linux to implement email security.		10/10/24	
13	Written Assignment - I		3/9/24	
14	Written Assignment - II		7/9/24	

Name: Saikarthik Iyer

Batch: T13

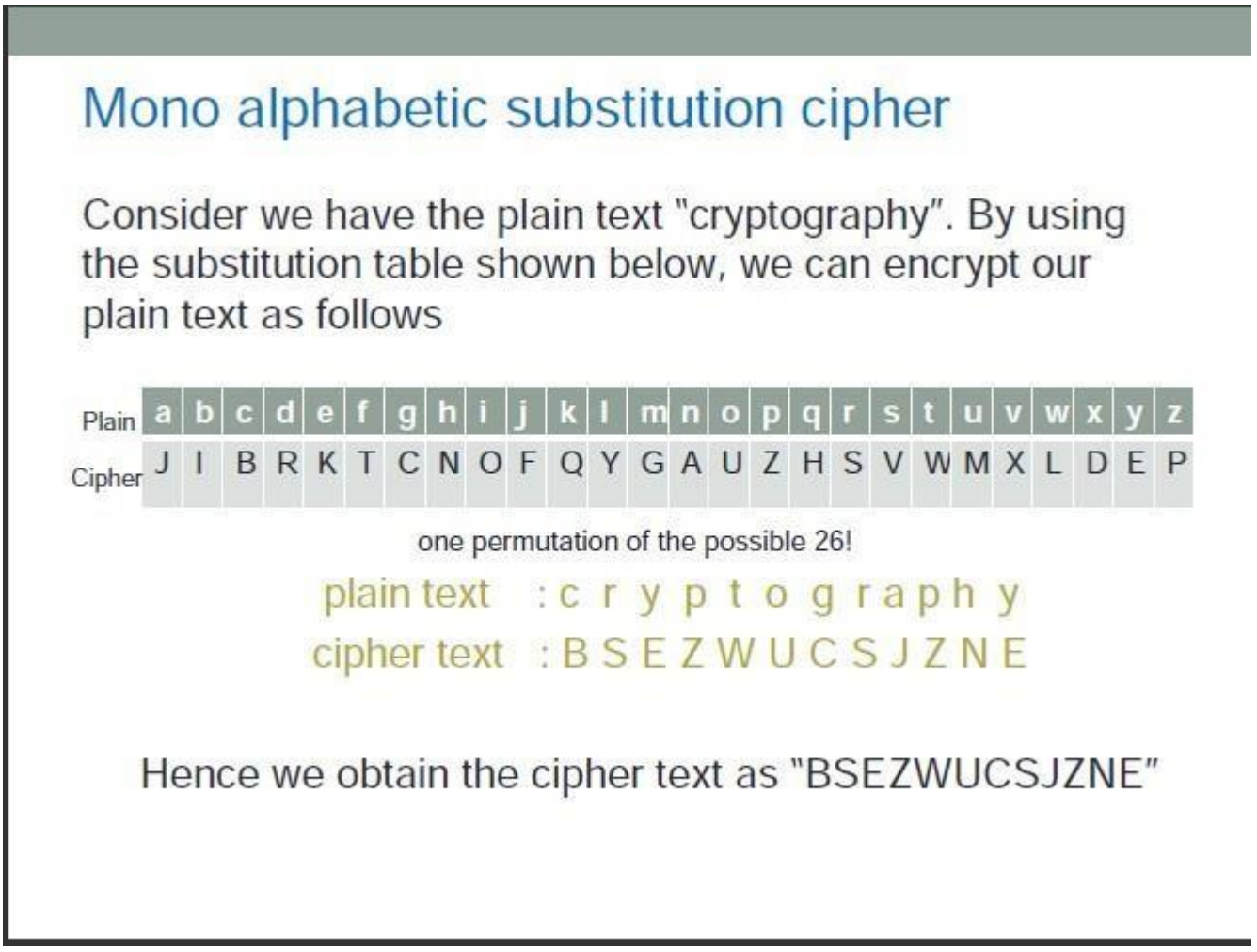
Roll No: 41

Assignment No. 1

Aim: Shift cipher and mono alphabet substitution cipher.

Theory:

Monoalphabetic Cipher is a part of the substitution technique in which a single cipher alphabet is used per message (mapping is done from plain alphabet to cipher alphabet). Monoalphabetic cipher converts plain text into cipher text and re-convert a cipher text to plain text. Monoalphabetic Cipher eliminates the brute-force techniques for cryptanalysis. Moreover, the cipher line can be a permutation of the 26 alphabetic characters.



Implementation:

PART II

Do your rough work here:

PART III

Plaintext:

attack at dawn

shift: 7

v Encrypt v | ^ Decrypt ^

Ciphertext

haahjr ha khdu

PART IV

Enter your solution Plaintext and shift key here:

attack at dawn

Key 7

Check my answer!

CORRECT!!

PART III

Enter your solution plaintext here:

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

Solution Key = xcdqrlpkwzoufteyahnvisgjbm

Check Answer!

CORRECT!!

RABBIT WITH A POCKET WATCH RUN PAST. SHE FOLLOWS IT DOWN A RABBIT HOLE WHEN SUDDENLY SHE FALLS A LONG WAY TO A CURIOUS HALL WITH MANY LOCKED DOORS OF ALL SIZES. SHE FINDS A SMALL KEY TO A DOOR TOO SMALL FOR HER TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING.

Modify the text above (in scratchpad):

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

Replace cipher character [s] by plaintext character [V] Modify

Use the following function to undo any unwanted exchange by giving an uppercase character and a lower case. This is a case sensitive function:

Replace character [N] by character [g] Replace these exact characters

Your replacement history:

You replaced d by C You replaced k by H You replaced x by A You replaced y by P You replaced v by T You replaced r by E You replaced h by R You replaced q by D You replaced e by O You replaced g by N You replaced N by g You replaced g by W You replaced t by N You replaced c by B You replaced w by I You replaced u by L You replaced n by S You replaced p by G You replaced o by K You replaced i by U You replaced l by F You replaced m by Z You replaced f by M You replaced b by Y You replaced s by V

PART I

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

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

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

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

Next Ciphertext

Calculate Frequencies in ciphertext

Ciphertext Frequencies:

Saikarthik Iyer
T1341

Assignment no. 2

Aim: To implement Playfair and Vigenere cipher.

Theory:

Playfair Cipher

The Playfair Cipher is a digraph substitution cipher, meaning it encrypts pairs of letters instead of single letters. Here's how it works:

1. Create a 5x5 matrix using a keyword. For example, let's use the keyword `MONARCHY`. The letters `I` and `J` are usually combined to fit the 25-letter grid.

M O N A R

C H Y B D

E F G I K

L P Q S T

U V W X Z

- Fill in the keyword first, skipping duplicate letters.
- Then, fill in the remaining letters of the alphabet.

2. Encrypting a message: Let's encrypt the message `HELLO`.

- Pair the letters: `HE` `LL` `O`. If a pair has the same letter (like `LL`), insert an `X` between them: `HE` `LX` `LO`.
- For each pair, find the letters in the grid:
- `H` and `E`: They form a rectangle, so take the letters on the opposite corners: `HF` → `BM`.
- `L` and `X`: They form a rectangle, so take the letters on the opposite corners: `LP` → `SU`.
- `L` and `O`: They are in the same row, so take the letters to their right: `LO` → `P`. - The encrypted message is: **BM SU PX**

Saikarthik Iyer
T1341

Vigenère Cipher

The Vigenère Cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution.

1. Choose a keyword: Let's use `LEMON`.

2. Encrypt a message: Let's encrypt the message `ATTACKATDAWN`.

- Repeat the keyword to match the length of the message: `LEMONLEMONLE`.
- Align the plaintext with the keyword:

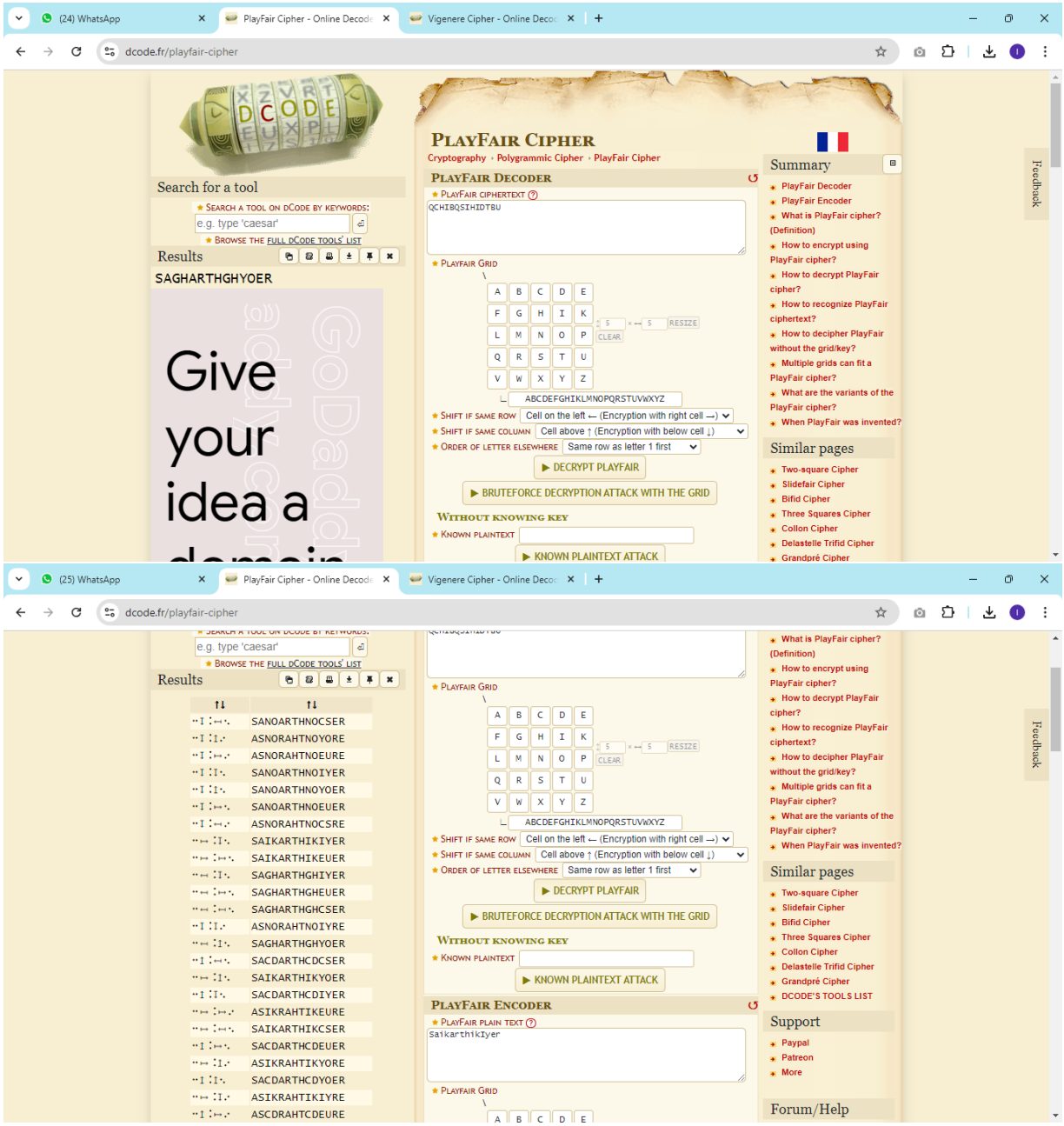
Plaintext: ATTACKATDAWN

Keyword: LEMONLEMONLE

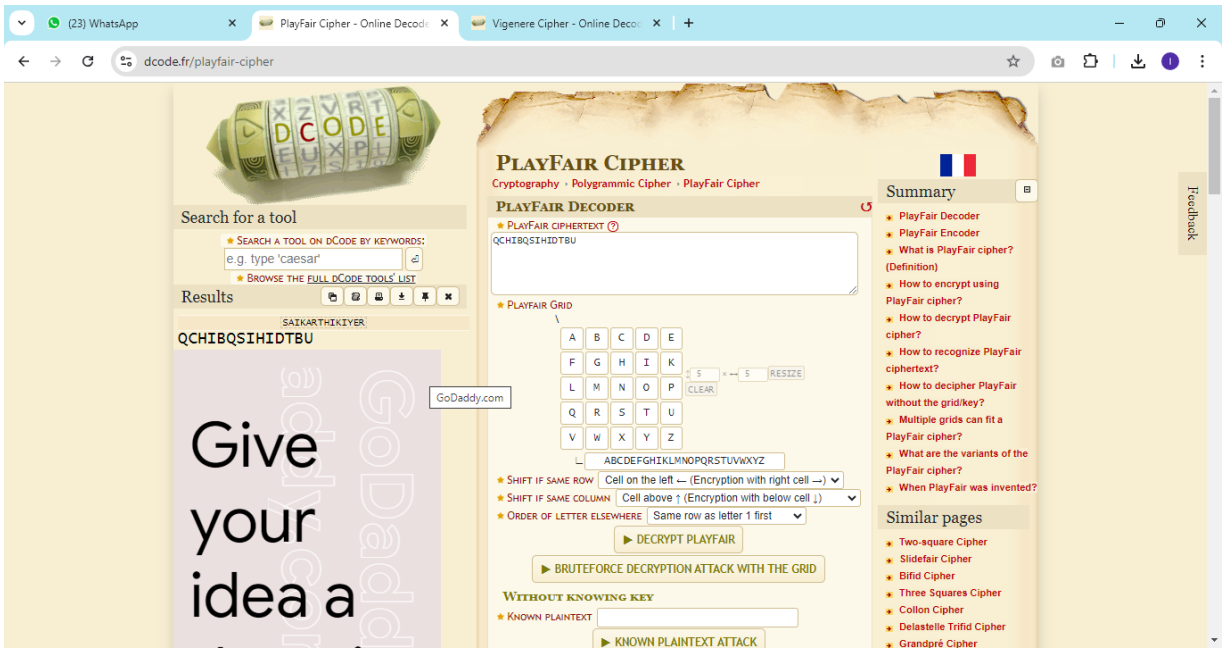
- For each letter in the plaintext:
- Shift it by the value of the corresponding letter in the keyword using the Vigenère table or by simple Caesar shift.
- `A` + `L` = `L`
- `T` + `E` = `X`
- `T` + `M` = `F`
- `A` + `O` = `O`
- `C` + `N` = `P`
- `K` + `L` = `V`
- `A` + `E` = `E`
- `T` + `M` = `F`
- `D` + `O` = `R`
- `A` + `N` = `N`
- `W` + `L` = `H`
- `N` + `E` = `R`

- The encrypted message is: LXFOPVEFRNHR.

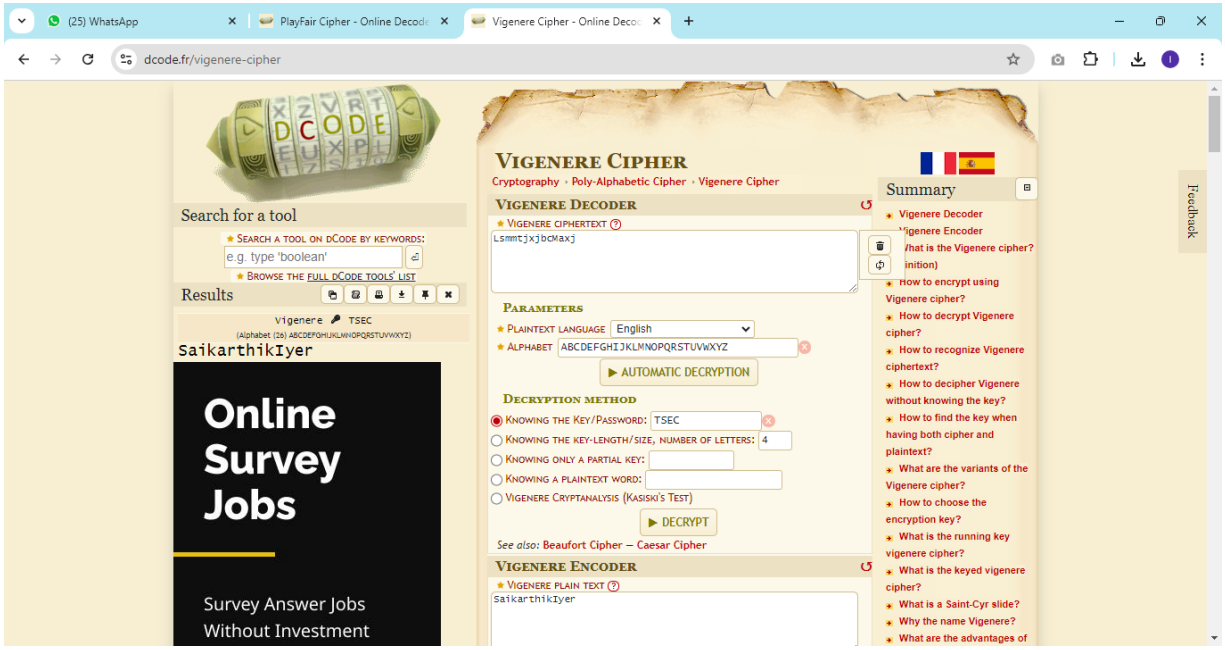
Saikarthik Iyer
T1341

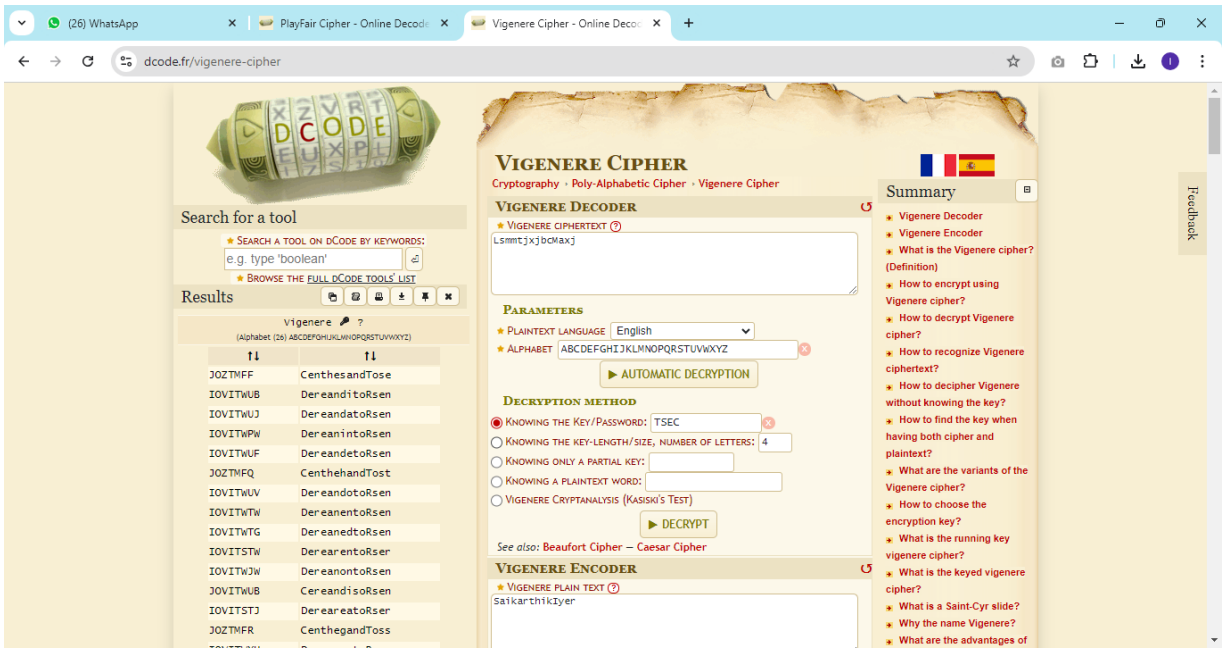


Saikarthik Iyer
T1341



Saikarthik Iyer
T1341





Name: Saikarthik Iyer

Batch: T1341

Subject: Security Lab-Assignment 3

Aim: **Block Cipher Modes of operation Theory:**

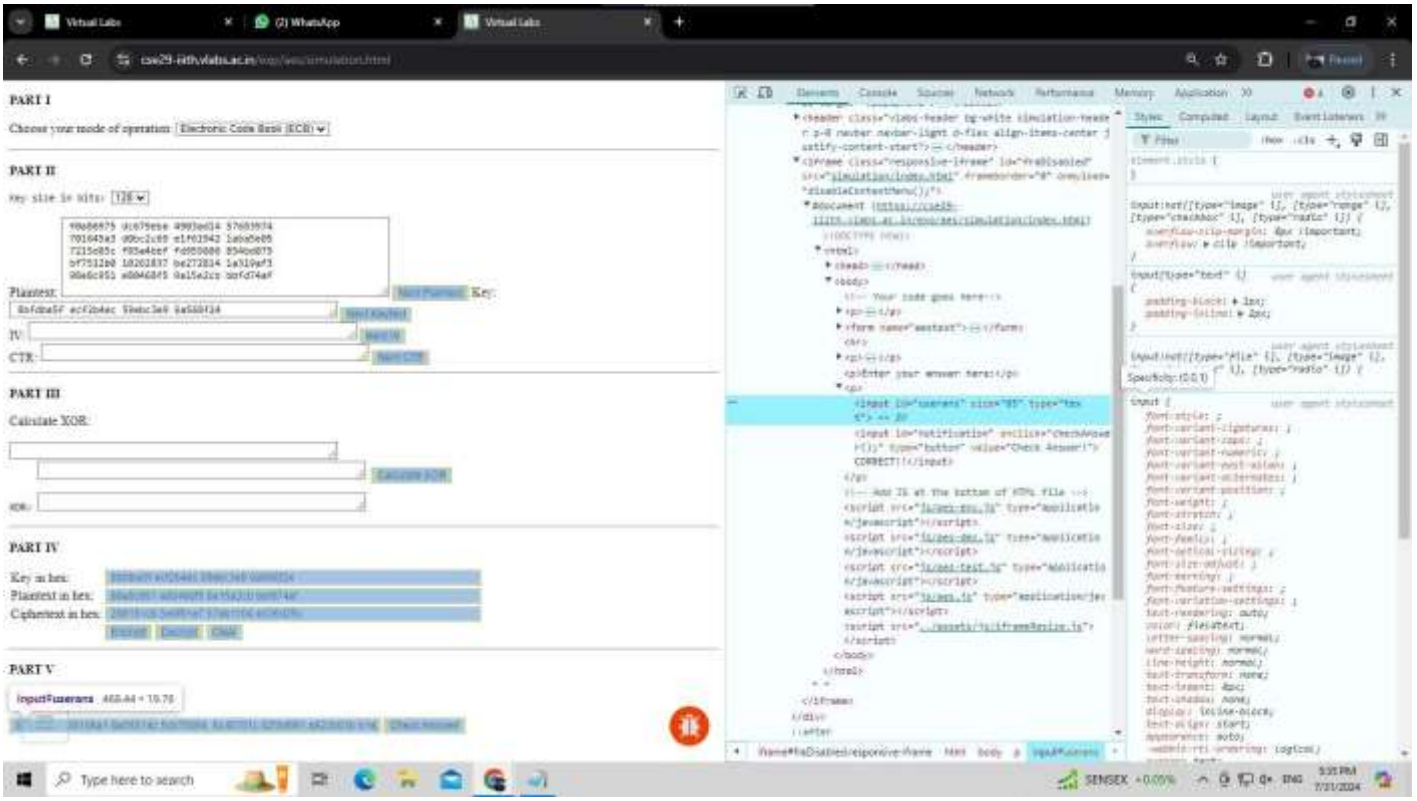
Common Block Cipher Modes of Operation

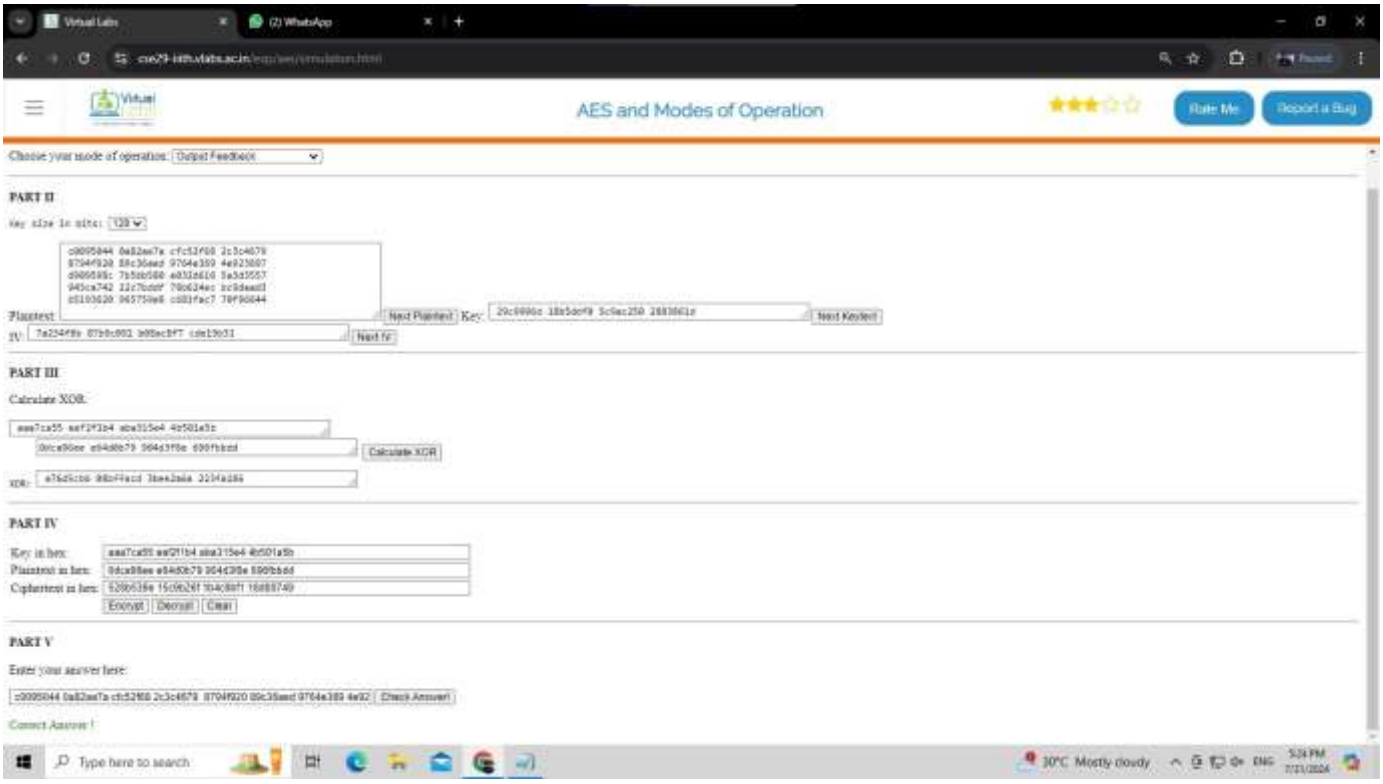
1. **Electronic Code Book (ECB):**
    - Simplest mode: Each plaintext block is encrypted independently.
    - Vulnerable to frequency analysis due to identical ciphertext blocks for identical plaintext blocks.
    - Rarely used in practice due to security concerns.
2. **Cipher Block Chaining (CBC):**
    - Each plaintext block is XORed with the previous ciphertext block before encryption.
    - Introduces dependency between blocks, improving security.
    - Requires an initialization vector (IV) for the first block.
3. **Cipher Feedback (CFB):**
    - Converts a block cipher into a stream cipher.
    - Previous ciphertext is encrypted, and the result is XORed with the plaintext to produce the ciphertext.
    - Similar to CBC but with feedback based on ciphertext.
4. **Output Feedback (OFB):**
    - Another stream cipher mode.
    - Generates a keystream by encrypting a counter.
    - Keystream is XORed with plaintext to produce ciphertext.
5. **Counter (CTR):**
    - Similar to OFB but uses a counter instead of feedback.
    - Provides high performance and can be parallelized.
    - Offers advantages in terms of error propagation and random access.

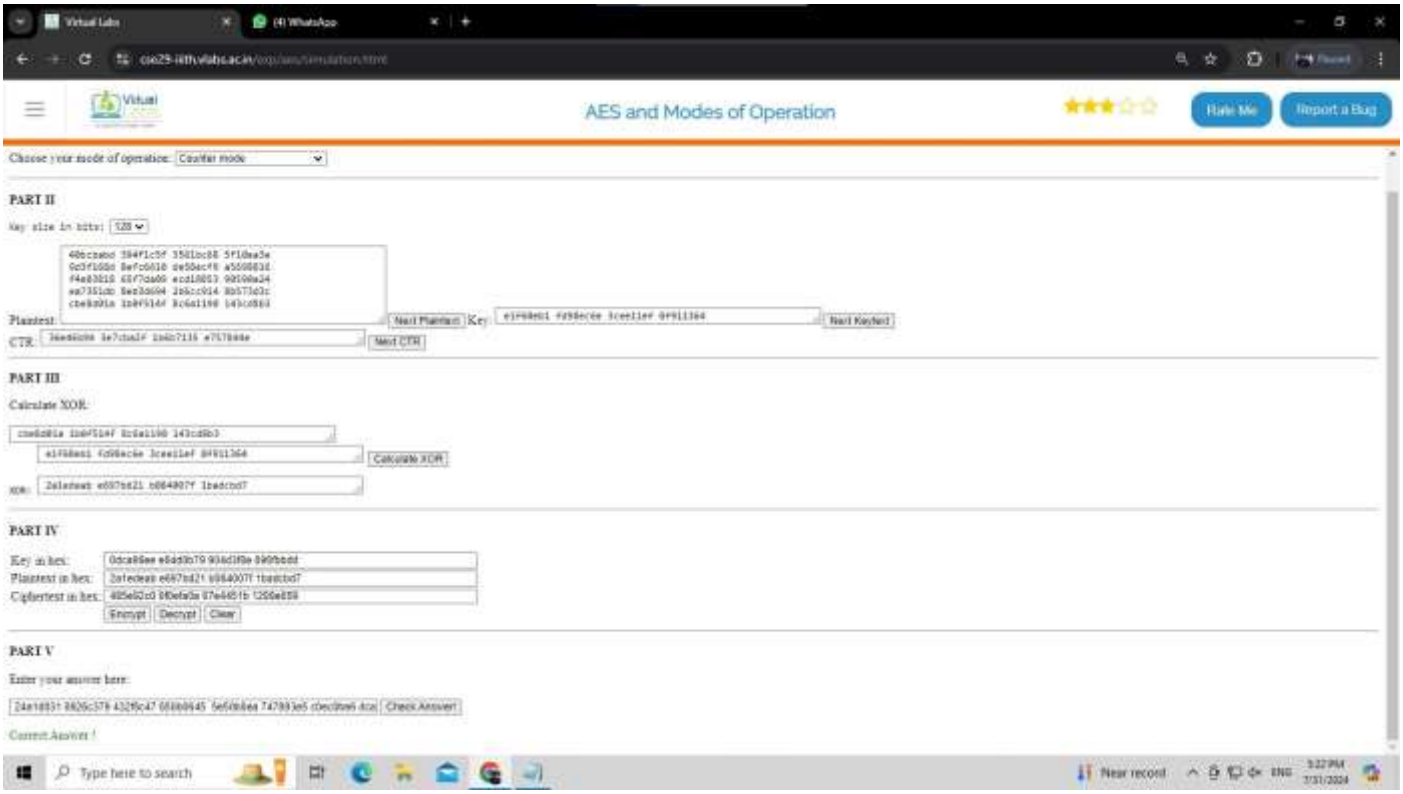
Key Considerations

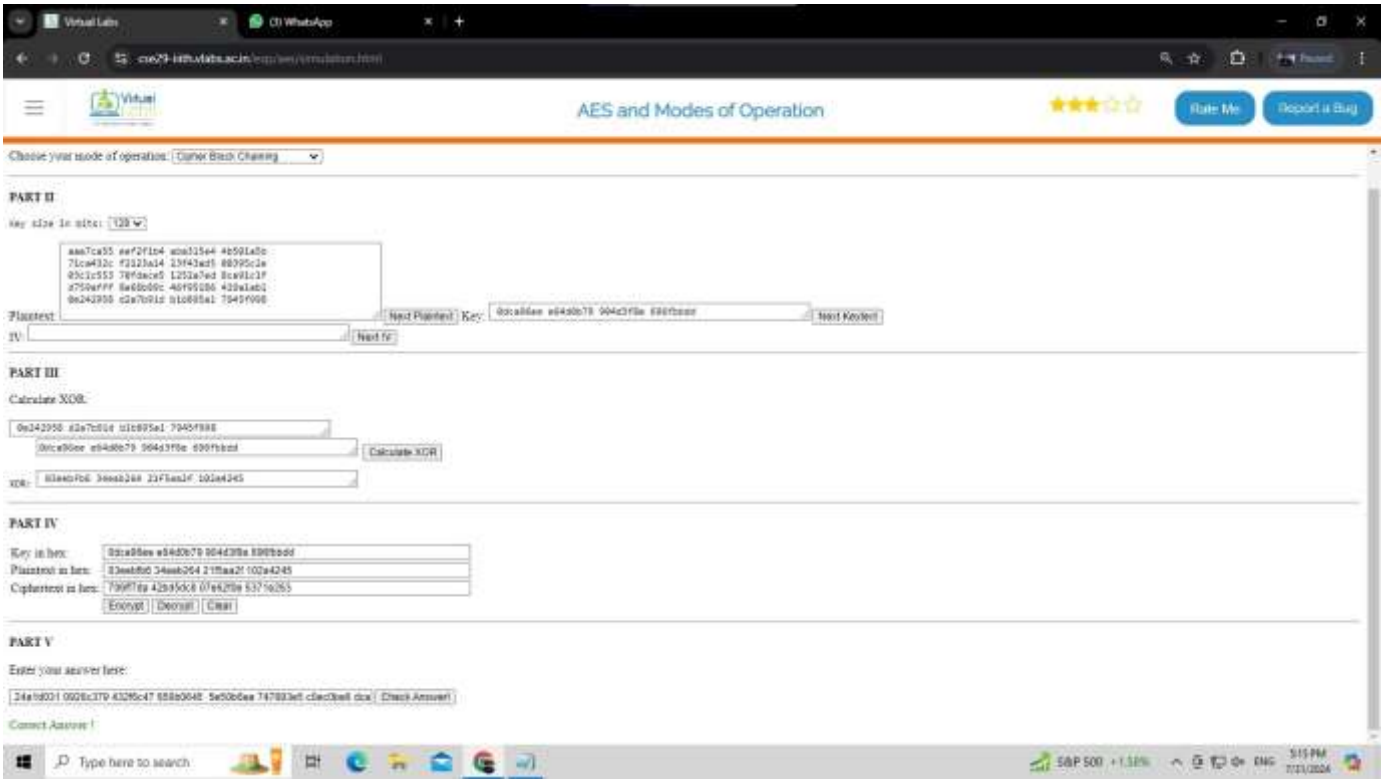
- **Security:** Different modes offer varying levels of security against attacks.
- **Performance:** Some modes are more efficient than others.
- **Error propagation:** Some modes are more resilient to bit errors.
- **Random access:** Some modes allow for random access to ciphertext blocks.

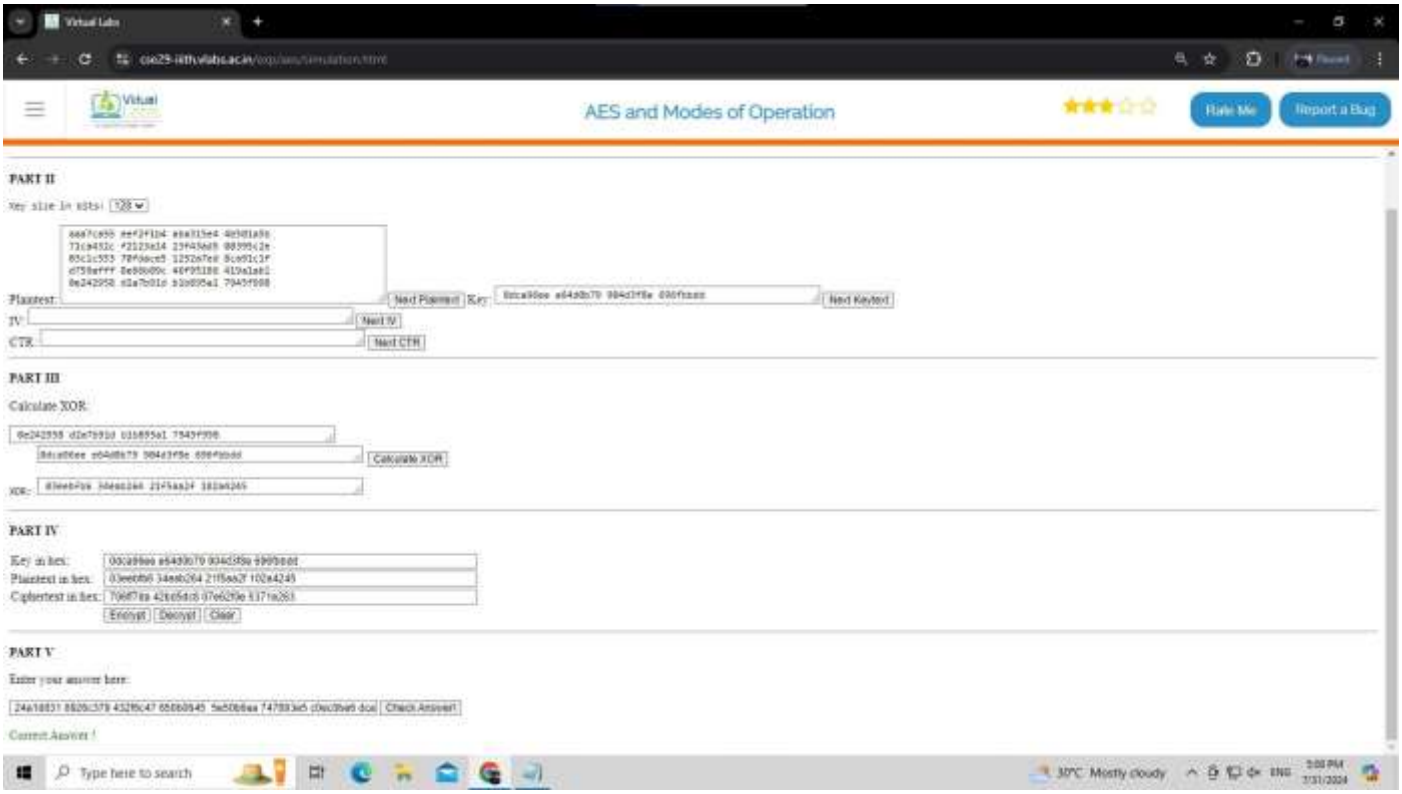
Implementation:






AES and Modes of Operation
Rate Me
Report a Bug
Choose your mode of operation:
PART II
PART III
Calculate XOR
Calculate XOR
PART IV
Key in hex:
Plaintext in hex:
Ciphertext in hex:
Encrypt
Decrypt
Clear
PART V
Enter your answer here
Check Answer
Correct Answer !


AES and Modes of Operation
Rate Me
Report a Bug
Choose your mode of operation: Cipher Block Chaining
PART II
Key size in bits: 128
Plaintext:
Next Plaintext
Key:
Next Keytext
IV:
Next IV
PART III
Calculate XOR:
Calculate XOR
XOR:
PART IV
Key in hex:
Plaintext in hex:
Ciphertext in hex:
Encrypt
Decrypt
Clear
PART V
Enter your answer here:
Check Answer!
Correct Answer !
Type here to search
S&P 500 +1.16%
ENG
5:15 PM

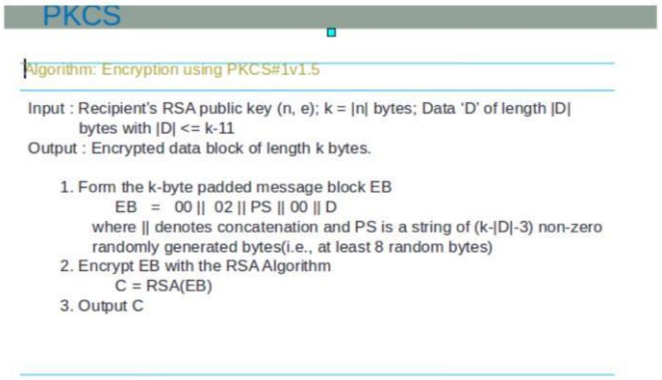

Virtual Labs
AES and Modes of Operation
Rate Me
Report a Bug
PART II
Key size in bits:
Plaintext:
Next Plaintext
Key:
Next Keytext
IV:
Next IV
CTR:
Next CTR
PART III
Calculate XOR
Calculate XOR
XOR:
PART IV
Key in hex:
Plaintext in hex:
Ciphertext in hex:
Encrypt
Decrypt
Clear
PART V
Enter your answer here
Check Answer!
Correct Answer !
Type here to search
30°C Mostly cloudy
ENG

Name: Saikarthik Iyer
Batch: T13
Roll no: 41

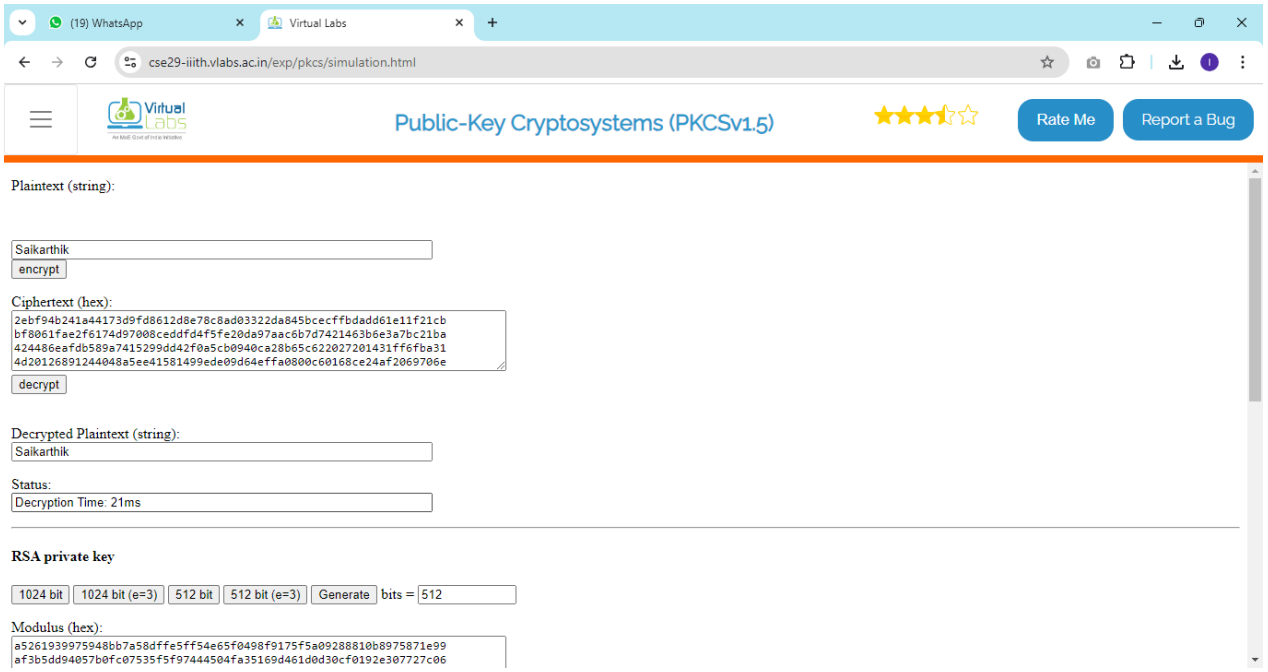
Assignment No. 4

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

Theory:



Output:



Conclusion: In conclusion, the implementation and analysis of the RSA cryptosystem and Digital Signature scheme using RSA demonstrate the effectiveness of RSA in providing both encryption and authentication. The RSA cryptosystem ensures secure data transmission through public-key encryption, while the Digital Signature scheme adds an additional layer of security by enabling message integrity and non-repudiation. The combination of these two mechanisms highlights RSA's strength in securing sensitive communications, although it requires careful key

management and computational resources due to the large prime numbers involved in key generation.

LO Mapped: LO2

Name: Saikarthik Iyer
Batch: T13
Roll no: 41

Assignment No 5

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

WHOIS:
Whois is a command-line utility used in Linux systems to retrieve information about domain names, IP addresses, and network devices registered with the Internet Corporation for Assigned Names and Numbers (ICANN). The data received by Whois consists of the name and contact information of the domain or IP address owner, the registration and expiration date, the domain registrar, and the server information.

Name: Saikarthik Iyer
Batch: T13
Roll no: 41

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo apt install whois
[sudo] password for lab1006:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  whois
0 upgraded, 1 newly installed, 0 to remove and 568 not upgraded.
Need to get 43.7 kB of archives.
After this operation, 262 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 whois amd64 5.3.0 [43.7 kB]
Fetched 43.7 kB in 1s (86.5 kB/s)
Selecting previously unselected package whois.
(Reading database ... 162475 files and directories currently installed.)
Preparing to unpack .../archives/whois_5.3.0_amd64.deb ...
Unpacking whois (5.3.0) ...
Setting up whois (5.3.0) ...
Processing triggers for man-db (2.8.3-2) ...
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ whois tsec.edu
This Registry database contains ONLY .EDU domains.
The data in the EDUCAUSE Whois database is provided
by EDUCAUSE for information purposes in order to
assist in the process of obtaining information about
or related to .edu domain registration records.

The EDUCAUSE Whois database is authoritative for the
.EDU domain.

A Web interface for the .EDU EDUCAUSE Whois Server is
available at: http://whois.educause.edu

By submitting a Whois query, you agree that this information
will not be used to allow, enable, or otherwise support
the transmission of unsolicited commercial advertising or
solicitations via e-mail.  The use of electronic processes to
harvest information from this server is generally prohibited
except as reasonably necessary to register or modify .edu
domain names.

-----------------------------------------------------------
```

```
Domain Name: TSEC.EDU

Registrant:
        Thadomal Sahani Engineering College
        P.G Kher Marg, Bandra(W)
        Mumbai, Maharashtra 400 050
        India

Administrative Contact:
        Dr. Gopakumaran Thampi
        Thadomal Shahani Engineering College
        Nari Gurshahani Marg, Bandra(W)
        Mumbai, 400050
        India
        +91.2226495808
        gtthampi@yahoo.com

Technical Contact:
        Chetan Agarwal
        Thadomal Shahani Engineering College
        Nari Gurshahani Marg, Bandra(W)
        Mumbai, 400050
        India
        +91.2226495808
        chetan.agarwal@thadomal.org

Name Servers:
        NS1.SALESUPP.IN
        NS2.SALESUPP.IN

Domain record activated:    22-Jan-2001
Domain record last updated: 25-Jun-2024
Domain expires:             31-Jul-2025
```

Dig:

dig command stands for Domain Information Groper. It is used for retrieving information about DNS name servers. It is basically used by network administrators. It is used for verifying and troubleshooting DNS problems and to perform DNS lookups. Dig command replaces older tools such as nslookup and the host.

Name: Saikarthik Iyer
Batch: T13
Roll no: 41

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

; <<>> DiG 9.11.3-1ubuntu1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62155
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.google.com.         58      IN      A       142.251.42.36

;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri Aug 16 13:20:54 IST 2024
;; MSG SIZE  rcvd: 59

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dig tsec.edu

; <<>> DiG 9.11.3-1ubuntu1-Ubuntu <<>> tsec.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44080
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
tsec.edu.               5353    IN      A       162.241.70.62

;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri Aug 16 13:25:23 IST 2024
;; MSG SIZE  rcvd: 53
```

Traceroute:

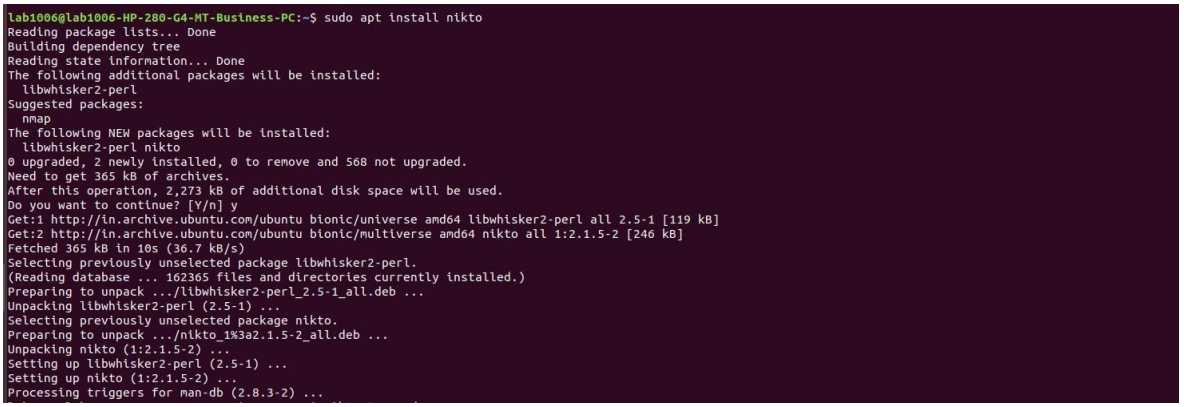
In networking, understanding the path that data packets take from one point to another is crucial for diagnosing and troubleshooting connectivity issues. One of the most valuable tools for this purpose is the traceroute command in Linux. Traceroute is a command-line tool used in Linux or other operating systems to track the path that data takes from your computer to a specified destination, such as a website.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute tsec.edu
traceroute to tsec.edu (162.241.70.62), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.459 ms  0.571 ms  0.344 ms
 2  203.212.25.1 (203.212.25.1)  1.871 ms  1.917 ms  1.919 ms
 3  203.212.24.53 (203.212.24.53)  2.153 ms  2.274 ms  2.113 ms
 4  175.100.177.53 (175.100.177.53)  3.448 ms  3.502 ms  3.381 ms
 5  172.16.2.101 (172.16.2.101)  6.217 ms  6.193 ms  3.353 ms
 6  121.241.42.57.static-mumbai.vsnl.net.in (121.241.43.57)  3.985 ms  4.036 ms  4.658 ms
 7  172.23.78.237 (172.23.78.237)  2.982 ms  2.941 ms *
 8  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net (180.87.38.5)  6.475 ms  6.085 ms  7.271 ms
 9  if-be-13-2.ecore1.mlv-mumbai.as6453.net (180.87.38.29)  128.436 ms  128.455 ms *
10  if-bundle-2-2.qcore2.mlv-mumbai.as6453.net (209.58.105.0)  125.228 ms * *
11  * if-bundle-29-2.qcore1.ldn-london.as6453.net (209.58.105.3)  128.404 ms  127.943 ms
12  * * *
13  be2868.ccr41.lon13.atlas.cogentco.com (154.54.57.153)  128.847 ms  129.475 ms  129.548 ms
14  be2317.ccr41.jfk02.atlas.cogentco.com (154.54.30.185)  194.478 ms be2490.ccr42.jfk02.atlas.cogentco.com (154.54.42.85)  198.003 ms  197.730 ms
15  be5181.ccr42.dca01.atlas.cogentco.com (154.54.165.17)  200.269 ms be4943.ccr41.dca01.atlas.cogentco.com (154.54.165.13)  200.141 ms be5181.ccr42.dca01.atlas.cogentco.com (154.54.165.17)  200.172 ms
16  be2112.ccr41.atl01.atlas.cogentco.com (154.54.7.158)  220.088 ms  219.608 ms be2113.ccr42.atl01.atlas.cogentco.com (154.54.24.222)  214.686 ms
17  be2127.rcr51.b009789-2.atl01.atlas.cogentco.com (154.54.82.210)  220.688 ms  222.182 ms be2126.rcr51.b009789-2.atl01.atlas.cogentco.com (154.54.82.206)  215.331 ms
18  38.104.183.2 (38.104.183.2)  223.669 ms  223.597 ms  223.574 ms
19  50-6-131-0.unifiedlayer.com (50.6.131.0)  214.938 ms  216.120 ms  215.656 ms
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

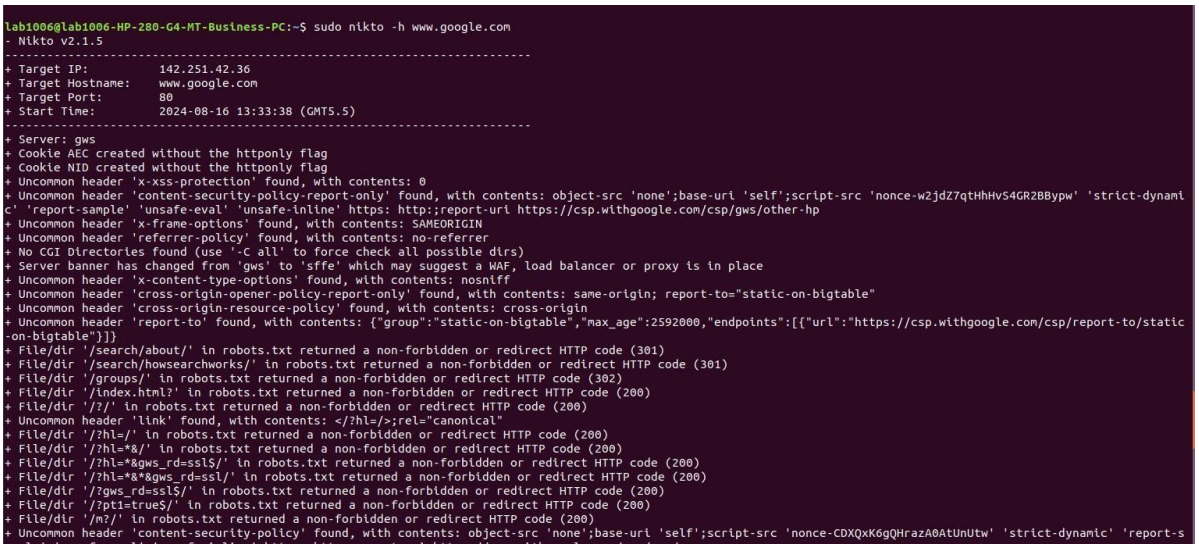
Nikto:

Name: Saikarthik Iyer
Batch: T13
Roll no: 41

Nikto is an open-source web vulnerability scanner based on Perl. It can scan for insecure files and programs, software misconfigurations, and other potential threats within the server. In this article, you will learn how to install and use Nikto to scan your Ubuntu server.



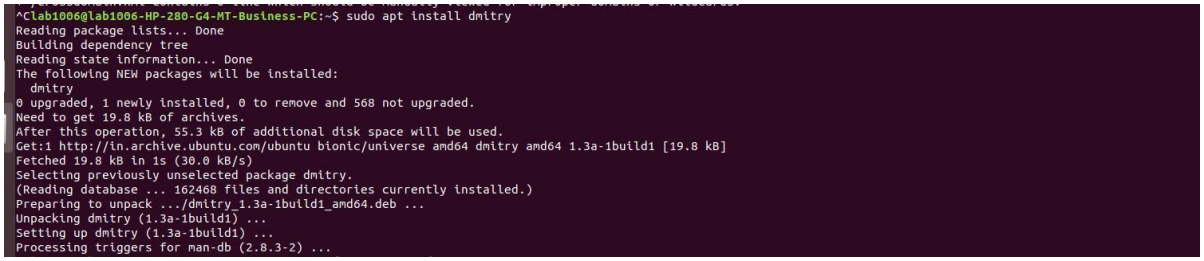
Name: Simran Yelave
Batch: T23
Roll no: 102

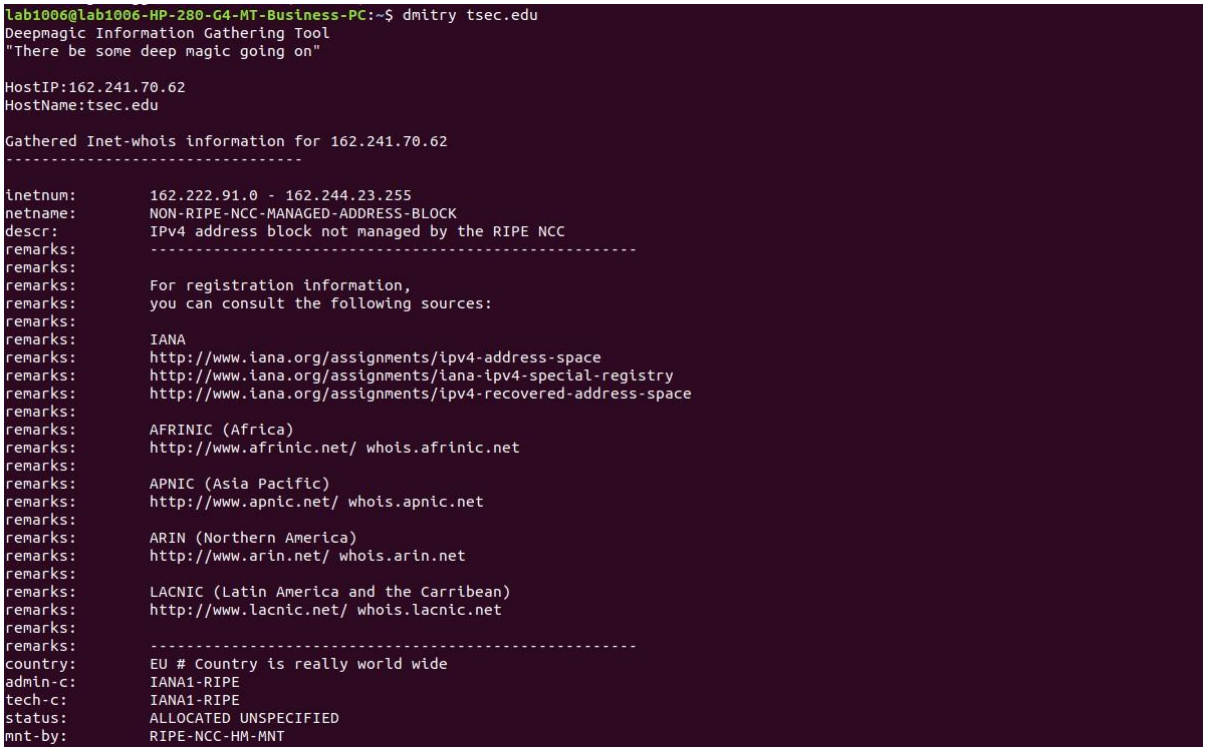


Name: Simran Yelave
Batch: T23
Roll no: 102

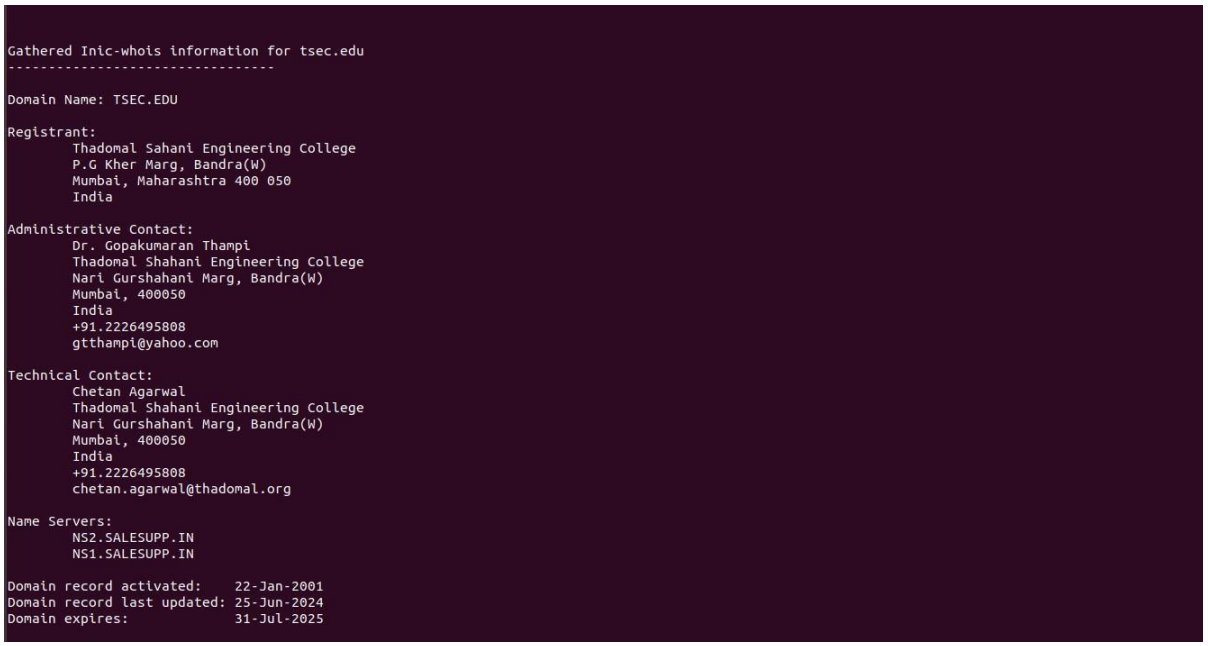
Dimitry:

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

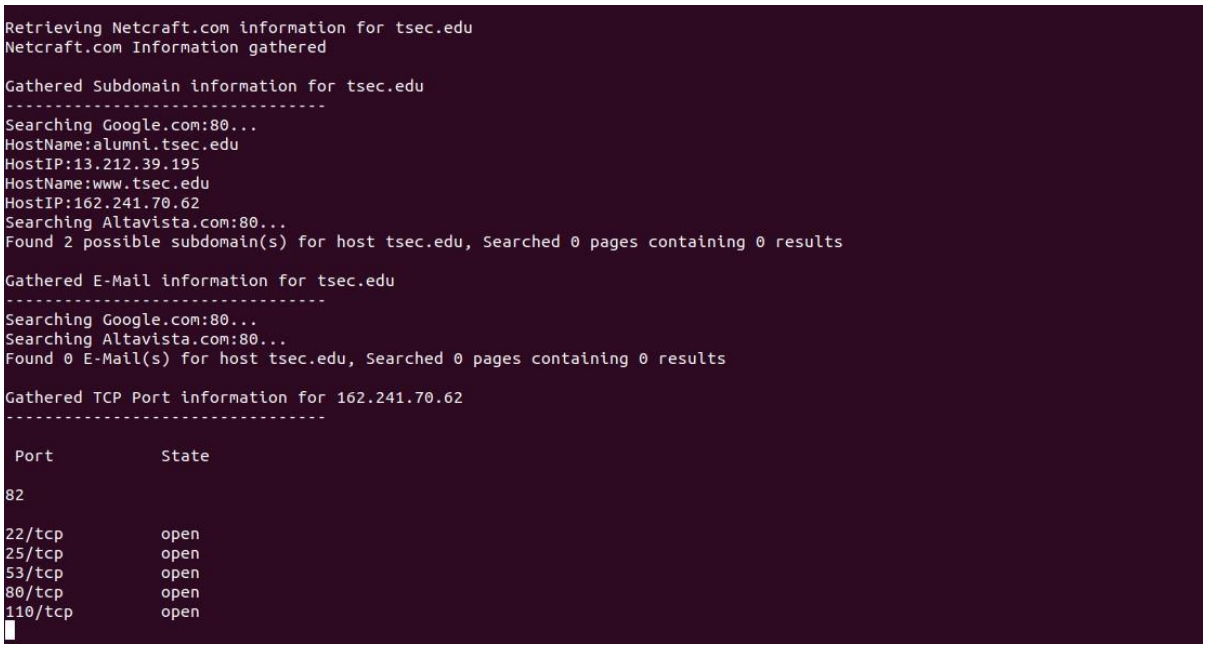




Name: Simran Yelave
Batch: T23
Roll no: 102



```
Gathered Inic-whois information for tsec.edu
--------------------------------

Domain Name: TSEC.EDU

Registrant:
        Thadomal Sahani Engineering College
        P.G Kher Marg, Bandra(W)
        Mumbai, Maharashtra 400 050
        India

Administrative Contact:
        Dr. Gopakumaran Thampi
        Thadomal Shahani Engineering College
        Nari Gurshahani Marg, Bandra(W)
        Mumbai, 400050
        India
        +91.2226495808
        gtthampi@yahoo.com

Technical Contact:
        Chetan Agarwal
        Thadomal Shahani Engineering College
        Nari Gurshahani Marg, Bandra(W)
        Mumbai, 400050
        India
        +91.2226495808
        chetan.agarwal@thadomal.org

Name Servers:
        NS2.SALESUPP.IN
        NS1.SALESUPP.IN

Domain record activated:    22-Jan-2001
Domain record last updated: 25-Jun-2024
Domain expires:             31-Jul-2025
```



```
Retrieving Netcraft.com information for tsec.edu
Netcraft.com Information gathered

Gathered Subdomain information for tsec.edu
---------------------------------
Searching Google.com:80...
HostName:alumni.tsec.edu
HostIP:13.212.39.195
HostName:www.tsec.edu
HostIP:162.241.70.62
Searching Altavista.com:80...
Found 2 possible subdomain(s) for host tsec.edu, Searched 0 pages containing 0 results

Gathered E-Mail information for tsec.edu
---------------------------------
Searching Google.com:80...
Searching Altavista.com:80...
Found 0 E-Mail(s) for host tsec.edu, Searched 0 pages containing 0 results

Gathered TCP Port information for 162.241.70.62
---------------------------------

 Port           State

82

22/tcp          open
25/tcp          open
53/tcp          open
80/tcp          open
110/tcp         open
```

Conclusion: We have seen commands are integral to network reconnaissance and troubleshooting. They provide valuable insights into domain ownership, DNS configurations, network paths, and potential vulnerabilities. Understanding and effectively utilizing these tools is essential for network administrators and security professionals in their efforts to maintain and secure network infrastructures.

Name: Simran Yelave
Batch: T23
LO Mapped: LO3

Saikarthik Iyer
T1341

LAB ASSIGNMENT NO.6

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

LABOUTCOMEATTAINED:

LO2: Demonstrate Key management, distribution and user authentication.

THEORY:

Hashing serves the crucial purpose of ensuring data integrity, security, and efficient data retrieval. It's

used in various applications like password storage, digital signatures, data verification, and more.

Hashing generates a fixed-size output (hash value) from an input (data), making it efficient for

comparing large datasets and detecting changes.

Different Hashing Algorithms:

1. MD5 (Message Digest Algorithm 5)

2. SHA-1 (Secure Hash Algorithm 1)

3. SHA-256 (Secure Hash Algorithm 256)

4. SHA-512 (Secure Hash Algorithm 512)

5. SHA-3 (Secure Hash Algorithm 3)

6. Whirlpool

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

1. Check Hashdeep Version: `hashdeep-V`

2. Display Help: `hashdeep-h` or `hashdeep-hh`

3. Manual Page: `man hashdeep`

4. Manual Page for Specific Algorithm: `man md5deep`

5. Hash a File: `hashdeep filename`

6. Hash with Hidden Paths: `hashdeep-b filename`

7. Suppress Errors: `hashdeep-s filename`

Saikarthik Iyer
T1341

8. Multiple Hash Algorithms: `hashdeep-c md5,sha1,sha256,tiger filename`

9. Hash Multiple Files (MD5): `hashdeep-c md5 *.txt`

10. Hash Multiple Files (MD5 & SHA-1): `hashdeep-c md5,sha1 *.txt`

11. Hashing Block of Files: `hashdeep-c md5-p 100 example.txt`

12. Recursive Hashing: `hashdeep-c md5-r /home/user/myfiles`

13. Redirect Output: `md5deep *.txt > hashset.txt`

14. Matching Mode Output: `md5deep-m hashset.txt *`

15. Suppress System Messages: `md5deep-s-m hashset.txt *`

16. Display Negatively Matching Files: `md5deep-s-x hashset.txt *`

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

17. Create Hashset and Audit:- Create Hashset: `hashdeep-c md5,sha1,sha256-r /home/user/myfiles > hashset1.txt`- Audit Files: `hashdeep-a-r-k hashset1.txt /home/user/myfiles`

18. Audit with New File (Fails):- Create New File: `touch /home/user/myfiles/newfile.txt`- Audit Again: `hashdeep-a-r-k hashset1.txt /home/user/myfiles`

19. Check Failed Points (Verbose):- Audit with Verbose: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`

20. Audit After Moving File:- Move File: `mv /home/user/myfiles/example.txt /tmp`- Audit Again: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`

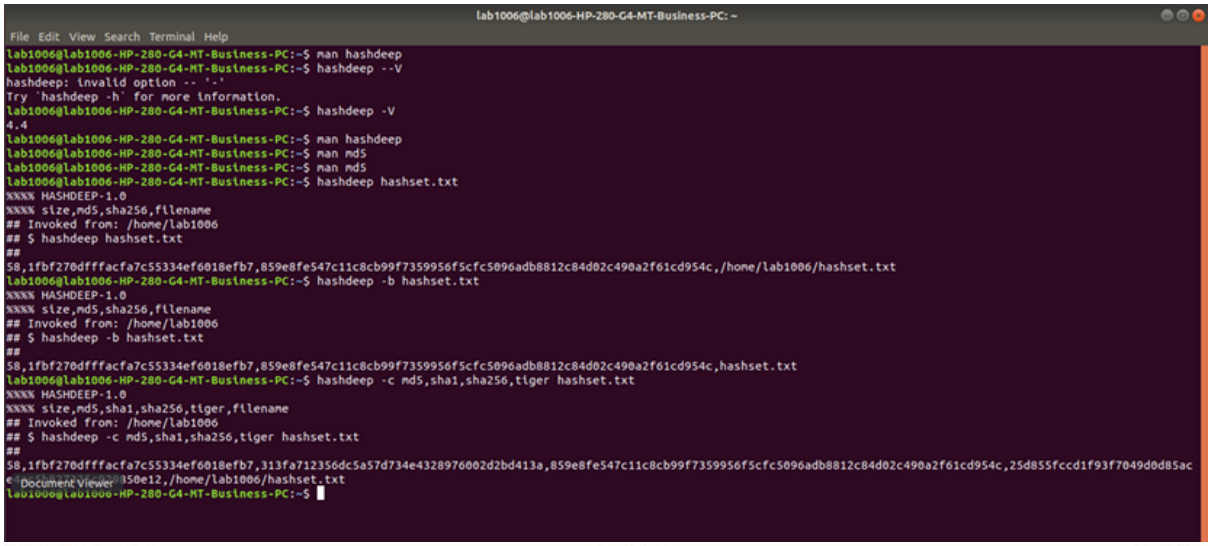
21. Audit After Renaming File:- Rename File: `mv /home/user/myfiles/shreya.txt /home/user/myfiles/backup.txt`- Audit Again: `hashdeep-v-a-r-k hashset1.txt /home/user/myfiles`

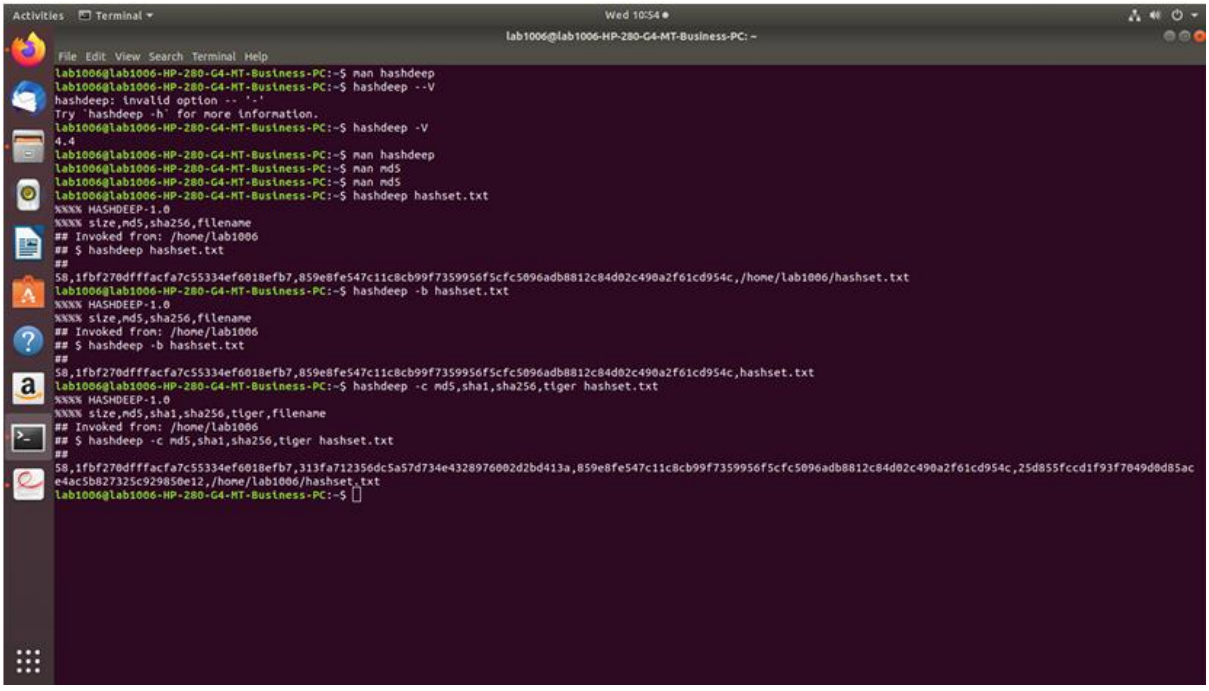
22. Verbose Audit Output:- More Verbose: `hashdeep-vv-a-r-k hashset1.txt /home/user/myfiles`- Very Verbose: `hashdeep-vvv-a-r-k hashset1.txt /home/user/myfiles`

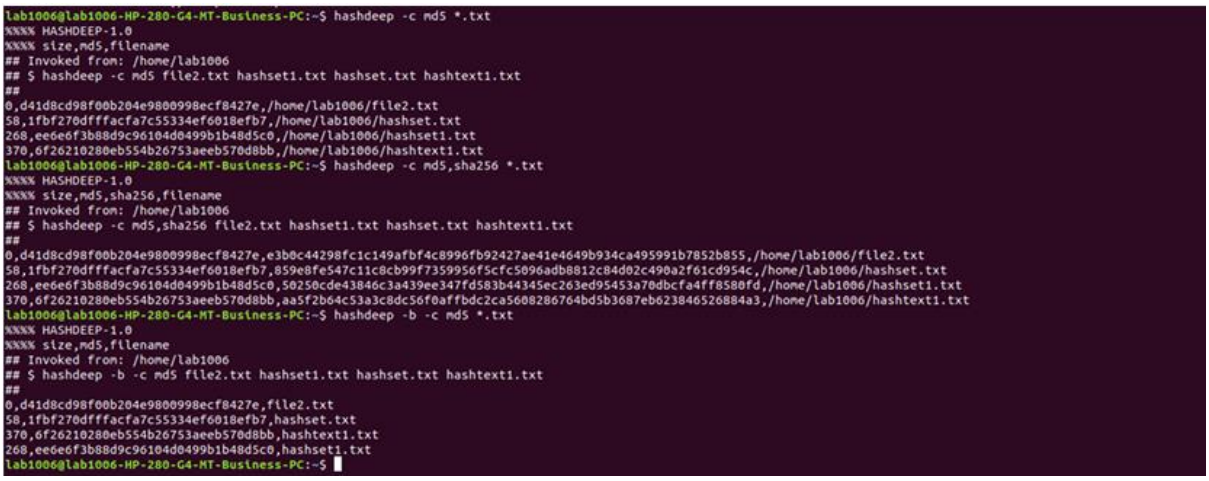
Note: Replace the paths and filenames with actual directory and file names as needed.

OUTPUT:

Saikarthik Iyer
T1341







Saikarthik Iyer
T1341

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$  hashdeep c md5 -r /home/lab1006/T13-53-CNS
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb6238465268 84a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS/1
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS/1
##
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS/2
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS/2
##
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb6238465268 84a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r /home/lab1006/T13-53-CNS
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -r c md5 /home/lab1006/T13-53-CNS
##
58,1fbf270dfffacfa7c55334ef6018efb7,859e8fe547c11c8cb99f7359956f5cfc5096adb8812c84d02c490a2f61cd954c,/home/lab1006/T13-53-CNS/1/hashset.txt
268,ee6e6f3b88d9c96104d0499b1b48d5c0,50250cde43846c3a439ee347fd583b44345ec263ed95453a70dbcfa4ff8580fd,/home/lab1006/T13-53-CNS/2/hashset1.txt
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/T13-53-CNS/1/file2.txt
370,6f26210280eb554b26753aeeb570d8bb,aa5f2b64c53a3c8dc56f0affbdc2ca5608286764bd5b3687eb6238465268 84a3,/home/lab1006/T13-53-CNS/2/hashtext1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep *.txt>hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset4.txt
d41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset4.txt
d41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep *.txt>hashset5.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset5.txt
9ed2bf26a0e00346cc78b1070d102897  /home/lab1006/hashset4.txt
ed5d34c74e59d16bd6d5b3683db655c3  /home/lab1006/file2.txt
d41d8cd98f00b204e9800998ecf8427e  /home/lab1006/hashoutput.txt
ee6e6f3b88d9c96104d0499b1b48d5c0  /home/lab1006/hashset1.txt
6f26210280eb554b26753aeeb570d8bb  /home/lab1006/hashtext1.txt
1fbf270dfffacfa7c55334ef6018efb7  /home/lab1006/hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

Name: Saikarthik Iyer
Batch:T13
Roll no:41

Assignment No. 7

Aim: Study of packet sniffer tools Wireshark and TCPDUMP.

Theory:

Wireshark and **TCPDump** are two widely used packet sniffer tools that capture and analyze network traffic. They are essential in network troubleshooting, security analysis, and protocol development.

Wireshark

Wireshark is a graphical packet analysis tool that provides a user-friendly interface for real-time traffic capture and detailed protocol inspection.

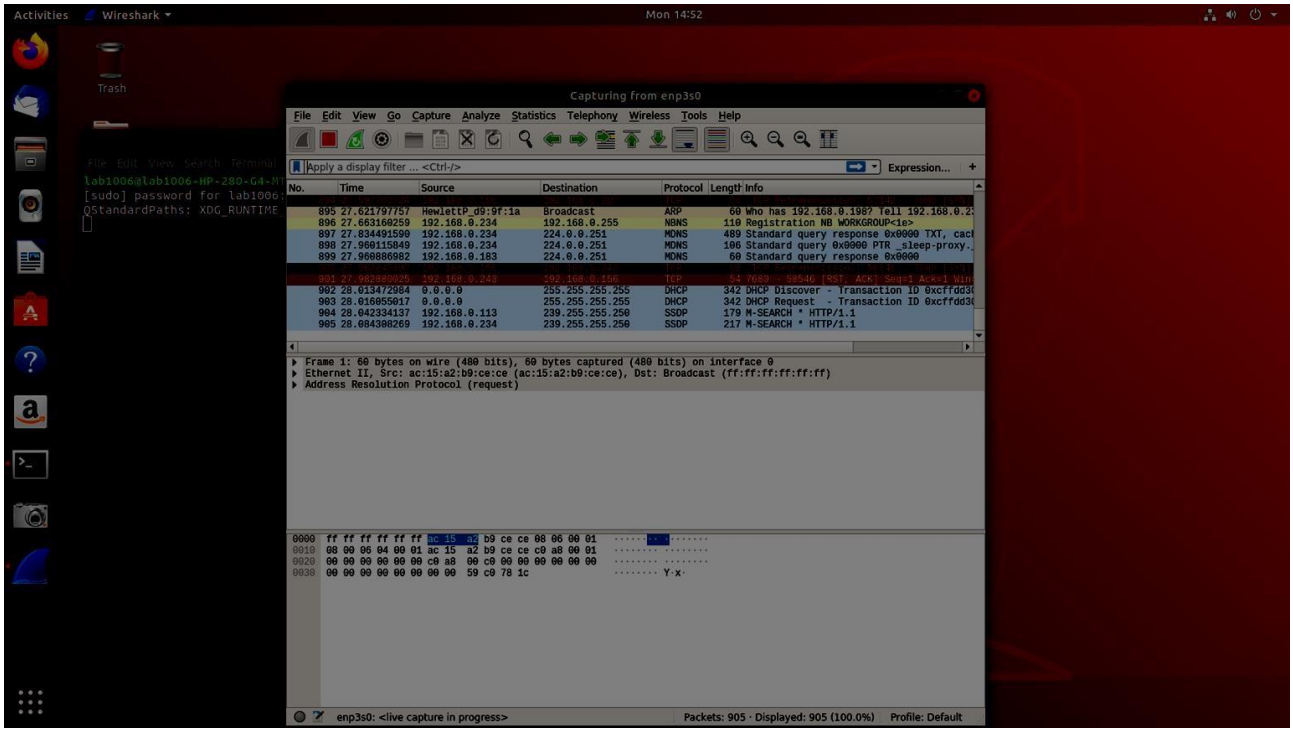
- **Features**: ○ Captures live traffic or imports packet capture files. ○ Provides detailed protocol analysis with color-coding for different types of traffic.
  - Supports a wide range of network protocols.
  - Allows filtering and searching within captured data for in-depth traffic analysis.
- **Use Cases**:
  - Visual analysis of network traffic. ○ Detecting network anomalies or misconfigurations. ○ Identifying protocol-specific issues in real-time.

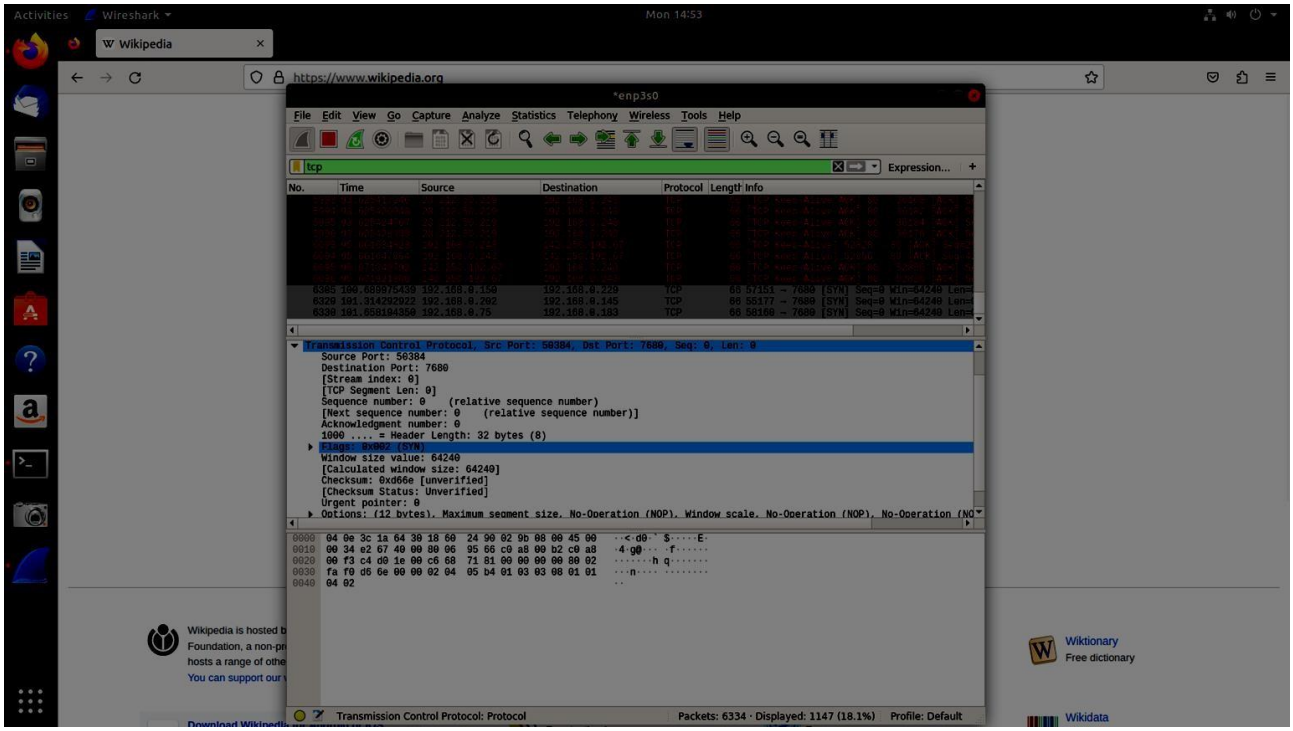
TCPDump

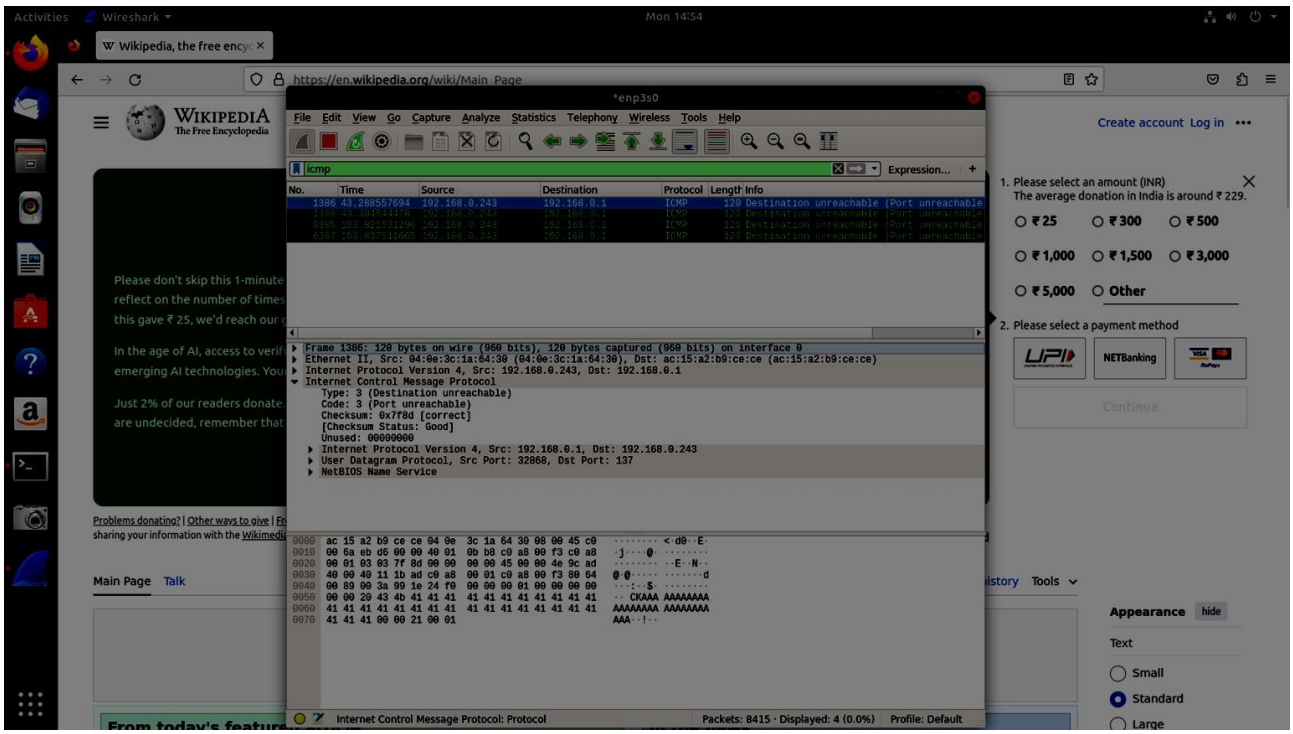
TCPDump is a command-line-based packet sniffer that captures network traffic at the data link layer and provides essential packet information.

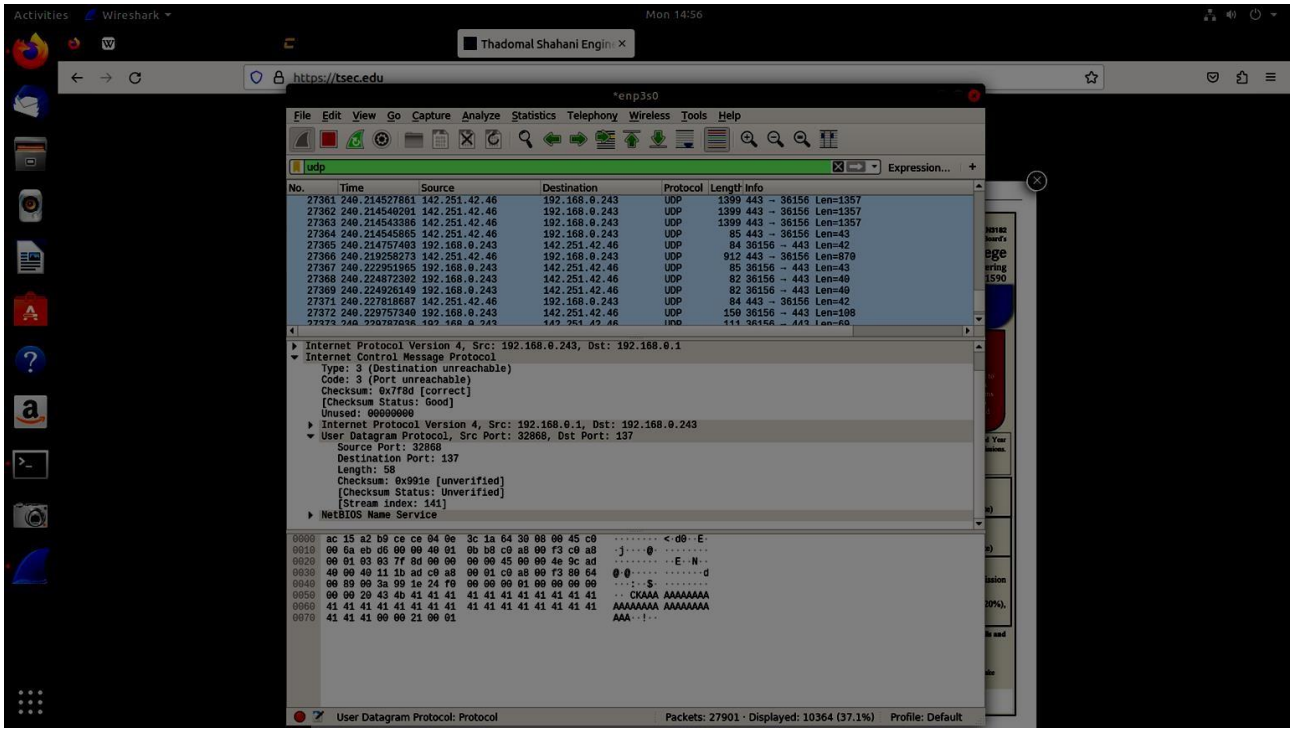
- **Features**:
  - Captures packets and displays them in the command line with a focus on simplicity and speed.
  - Offers extensive filtering options using the Berkeley Packet Filter (BPF) syntax.
  - Can save captured data to a file for later analysis with tools like Wireshark.
- **Use Cases**:
  - Lightweight packet capturing, especially in remote environments. ○ Filtering traffic based on IP addresses, protocols, or port numbers. ○ Scripting automated network traffic monitoring.

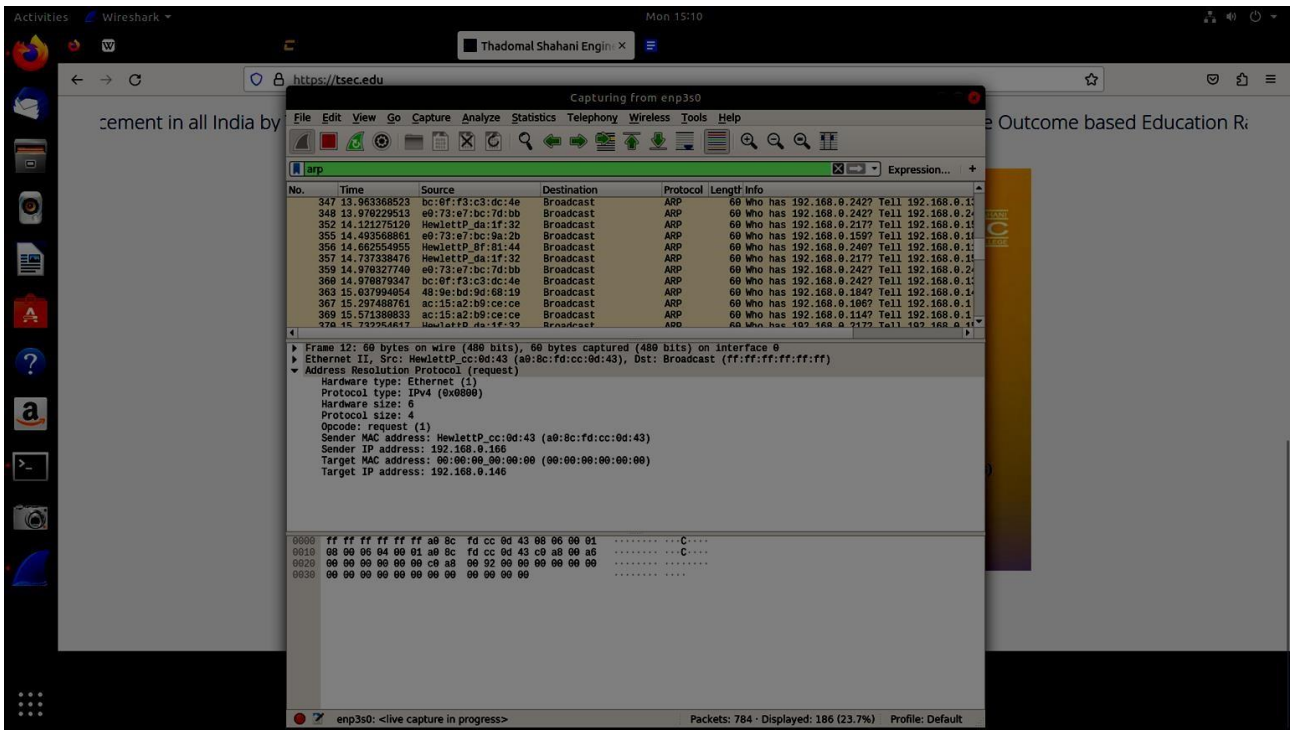
Output:

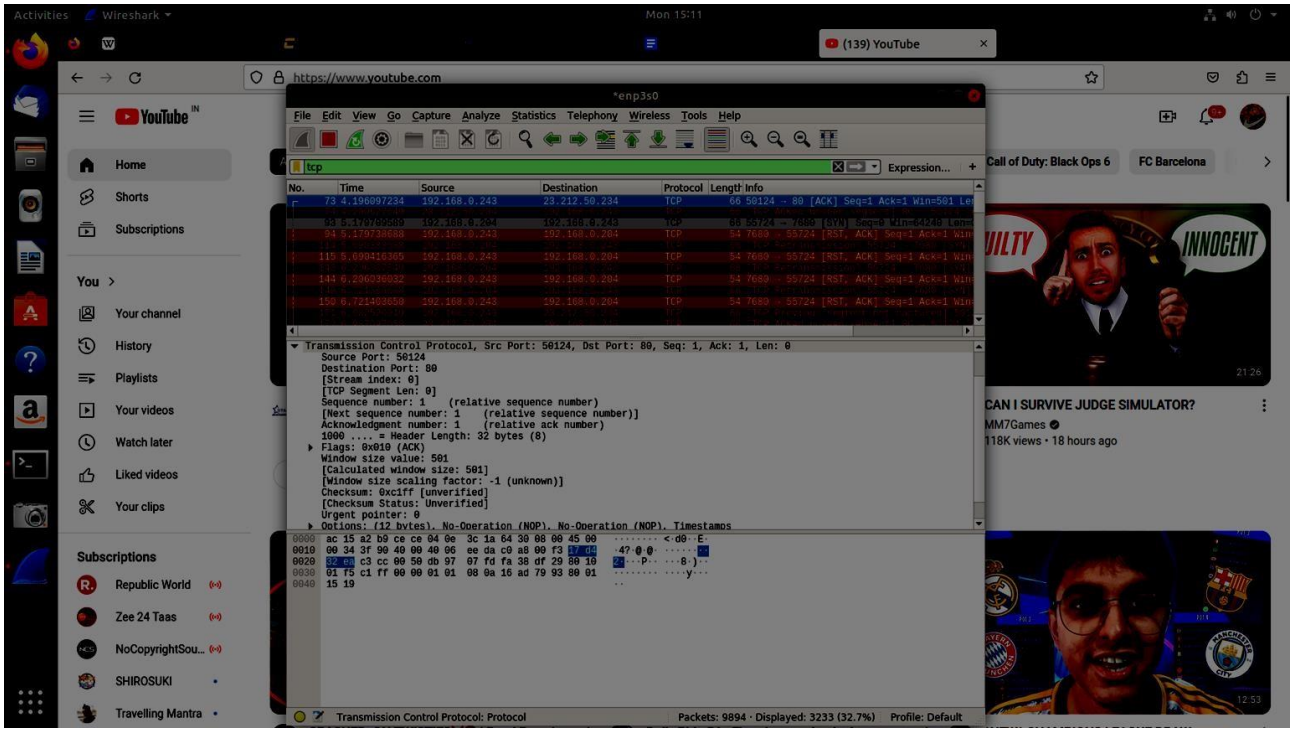

Activities
Wireshark
Mon 14:52
Trash
Capturing from enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
895 27.621797757 HewlettP_d9:9f:1a Broadcast ARP 60 Who has 192.168.0.198? Tell 192.168.0.2
896 27.663160259 192.168.0.234 192.168.0.255 NBNS 110 Registration NB WORKGROUP<1e>
897 27.834491590 192.168.0.234 224.0.0.251 MDNS 489 Standard query response 0x0000 TXT, cach
898 27.960115849 192.168.0.234 224.0.0.251 MDNS 106 Standard query 0x0000 PTR _sleep-proxy._
899 27.960886982 192.168.0.183 224.0.0.251 MDNS 60 Standard query response 0x0000
901 27.982240935 192.168.0.243 192.168.0.156 TCP 54 7680 → 58546 [RST, ACK] Seq=1 Ack=1 Win=
902 28.013472984 0.0.0.0 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xcffdd3
903 28.016055017 0.0.0.0 255.255.255.255 DHCP 342 DHCP Request - Transaction ID 0xcffdd3
904 28.042334137 192.168.0.113 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
905 28.084308269 192.168.0.234 239.255.255.250 SSDP 217 M-SEARCH * HTTP/1.1
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: ac:15:a2:b9:ce:ce (ac:15:a2:b9:ce:ce), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
0000 ff ff ff ff ff ff ac 15 a2 b9 ce ce 08 06 00 01
0010 08 00 06 04 00 01 ac 15 a2 b9 ce ce c0 a8 00 01
0020 00 00 00 00 00 00 c0 a8 00 c0 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 59 c0 78 1c
enp3s0: <live capture in progress>
Packets: 905 · Displayed: 905 (100.0%)
Profile: Default

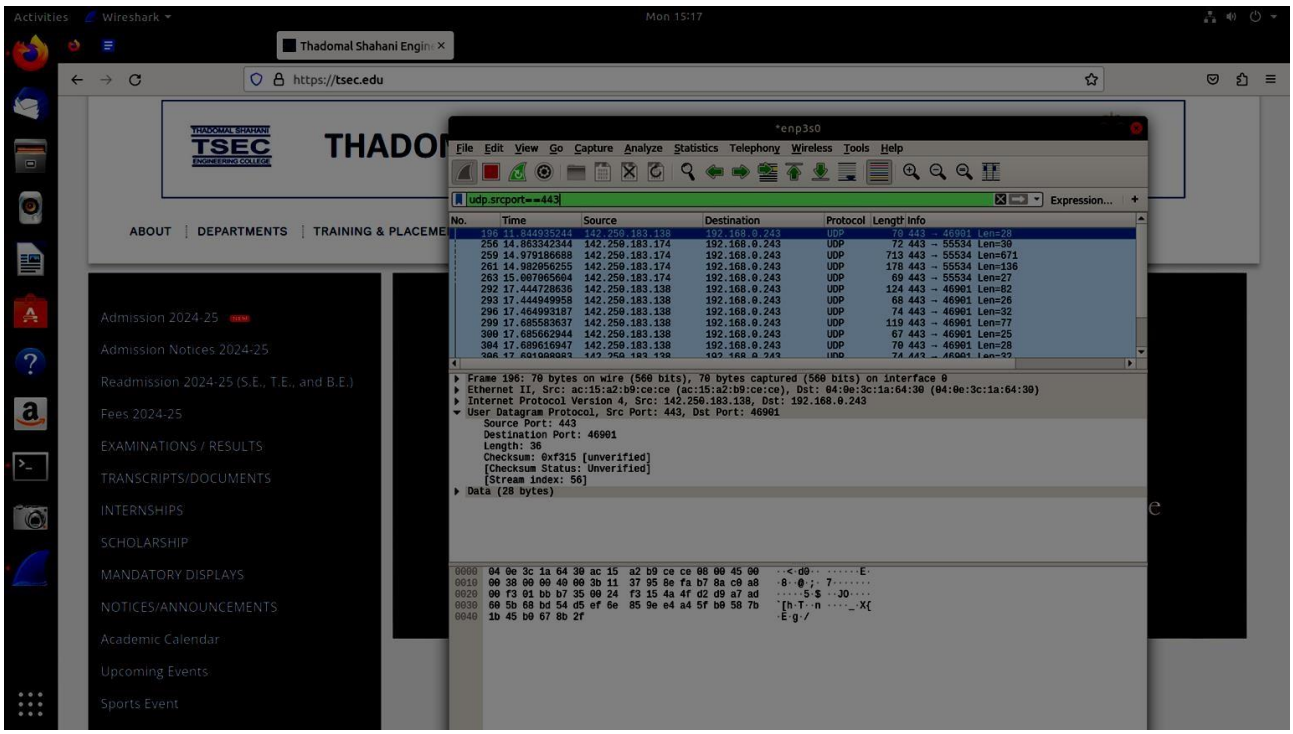

Activities
Wireshark
Mon 14:53
Wikipedia
https://www.wikipedia.org
*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
Expression...
No. Time Source Destination Protocol Length Info
6305 103.689875430 192.168.0.150 192.168.0.229 TCP 66 57151 → 7680 [SYN] Seq=0 Win=64240 Len=
6329 101.314292022 192.168.0.202 192.168.0.145 TCP 66 55177 → 7680 [SYN] Seq=0 Win=64240 Len=
6330 101.658194350 192.168.0.75 192.168.0.183 TCP 66 58160 → 7680 [SYN] Seq=0 Win=64240 Len=
Transmission Control Protocol, Src Port: 50384, Dst Port: 7680, Seq: 0, Len: 0
Source Port: 50384
Destination Port: 7680
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
Window size value: 64240
[Calculated window size: 64240]
Checksum: 0xd66e [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NO
0000 04 0e 3c 1a 64 30 18 60 24 90 02 9b 08 00 45 00
0010 00 34 e2 67 40 00 80 06 95 66 c0 a8 00 b2 c0 a8
0020 00 f3 c4 d0 1e 00 c6 68 71 81 00 00 00 00 80 02
0030 fa f0 d6 6e 00 00 02 04 05 b4 01 03 03 08 01 01
0040 04 02
Transmission Control Protocol: Protocol
Packets: 6334 · Displayed: 1147 (18.1%)
Profile: Default
Wikipedia is hosted b
Foundation, a non-pr
hosts a range of othe
You can support our
Wiktionary
Free dictionary
Wikidata

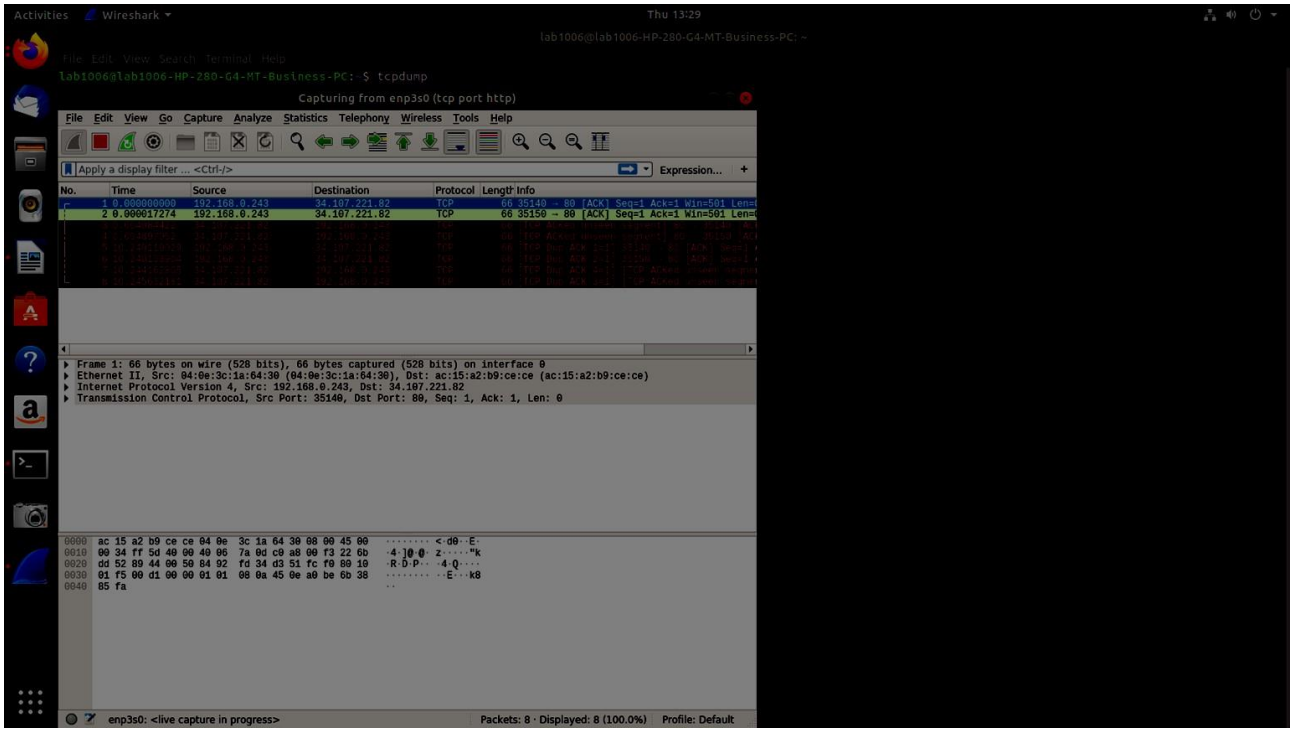

Activities
Wireshark
Mon 14:54
Wikipedia, the free ency
https://en.wikipedia.org/wiki/Main_Page
*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
icmp
Expression...
No. Time Source Destination Protocol Length Info
1386 43.288557694 192.168.0.243 192.168.0.1 ICMP 120 Destination unreachable (Port unreachable)
Frame 1386: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface 0
Ethernet II, Src: 04:0e:3c:1a:64:30 (04:0e:3c:1a:64:30), Dst: ac:15:a2:b9:ce:ce (ac:15:a2:b9:ce:ce)
Internet Protocol Version 4, Src: 192.168.0.243, Dst: 192.168.0.1
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
Checksum: 0x7f8d [correct]
[Checksum Status: Good]
Unused: 00000000
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.243
User Datagram Protocol, Src Port: 32868, Dst Port: 137
NetBIOS Name Service
Internet Control Message Protocol: Protocol
Packets: 8415 · Displayed: 4 (0.0%)
Profile: Default
Create account
Log in
1. Please select an amount (INR)
The average donation in India is around ₹ 229.
₹ 25
₹ 300
₹ 500
₹ 1,000
₹ 1,500
₹ 3,000
₹ 5,000
Other
2. Please select a payment method
UPI
NETBanking
Continue
Please don't skip this 1-minute
reflect on the number of times
this gave ₹ 25, we'd reach our
In the age of AI, access to verif
emerging AI technologies. You
Just 2% of our readers donate.
are undecided, remember that
Problems donating? | Other ways to give |
sharing your information with the Wikimedia
Main Page
Talk
Tools
Appearance
hide
Text
Small
Standard
Large

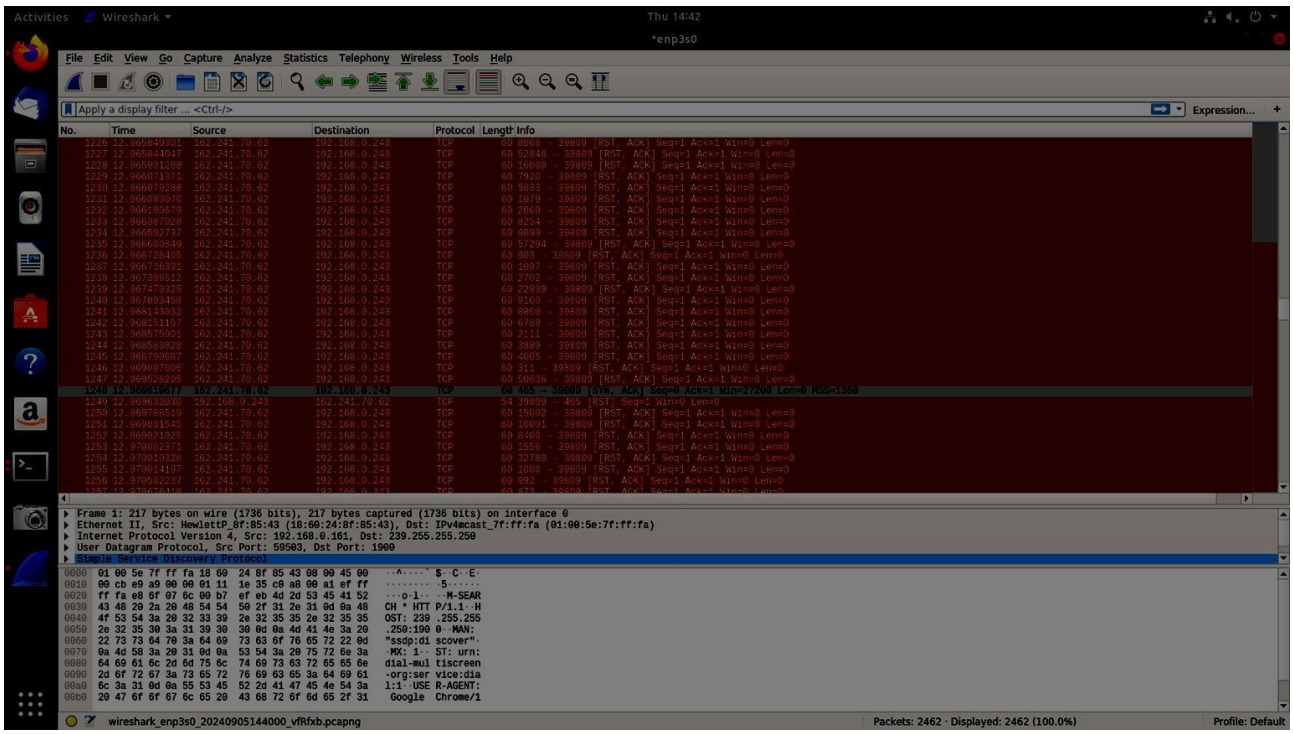

Activities
Wireshark
Mon 14:56
Thadomal Shahani Engin
https://tsec.edu
*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
Expression...
No. Time Source Destination Protocol Length Info
27361 240.214527861 142.251.42.46 192.168.0.243 UDP 1399 443 → 36156 Len=1357
27362 240.214540201 142.251.42.46 192.168.0.243 UDP 1399 443 → 36156 Len=1357
27363 240.214543386 142.251.42.46 192.168.0.243 UDP 1399 443 → 36156 Len=1357
27364 240.214545865 142.251.42.46 192.168.0.243 UDP 85 443 → 36156 Len=43
27365 240.214757403 192.168.0.243 142.251.42.46 UDP 84 36156 → 443 Len=42
27366 240.219258273 142.251.42.46 192.168.0.243 UDP 912 443 → 36156 Len=870
27367 240.222951965 192.168.0.243 142.251.42.46 UDP 85 36156 → 443 Len=43
27368 240.224872302 192.168.0.243 142.251.42.46 UDP 82 36156 → 443 Len=40
27369 240.224926149 192.168.0.243 142.251.42.46 UDP 82 36156 → 443 Len=40
27371 240.227818687 142.251.42.46 192.168.0.243 UDP 84 443 → 36156 Len=42
27372 240.229757340 192.168.0.243 142.251.42.46 UDP 150 36156 → 443 Len=108
Internet Protocol Version 4, Src: 192.168.0.243, Dst: 192.168.0.1
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
Checksum: 0x7f8d [correct]
[Checksum Status: Good]
Unused: 00000000
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.243
User Datagram Protocol, Src Port: 32868, Dst Port: 137
Source Port: 32868
Destination Port: 137
Length: 58
Checksum: 0x991e [unverified]
[Checksum Status: Unverified]
[Stream index: 141]
NetBIOS Name Service
User Datagram Protocol: Protocol
Packets: 27901 · Displayed: 10364 (37.1%)
Profile: Default


Activities
Wireshark
Mon 15:10
Thadomal Shahani Engin
https://tsec.edu
Capturing from enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp
Expression...
No. Time Source Destination Protocol Length Info
347 13.963368523 bc:0f:f3:c3:dc:4e Broadcast ARP 60 Who has 192.168.0.242? Tell 192.168.0.1
348 13.970229513 e0:73:e7:bc:7d:bb Broadcast ARP 60 Who has 192.168.0.242? Tell 192.168.0.2
352 14.121275120 HewlettP_da:1f:32 Broadcast ARP 60 Who has 192.168.0.217? Tell 192.168.0.1
355 14.493568861 e0:73:e7:bc:9a:2b Broadcast ARP 60 Who has 192.168.0.159? Tell 192.168.0.1
356 14.662554955 HewlettP_8f:81:44 Broadcast ARP 60 Who has 192.168.0.240? Tell 192.168.0.1
357 14.737338476 HewlettP_da:1f:32 Broadcast ARP 60 Who has 192.168.0.217? Tell 192.168.0.1
359 14.970327740 e0:73:e7:bc:7d:bb Broadcast ARP 60 Who has 192.168.0.242? Tell 192.168.0.2
360 14.970879347 bc:0f:f3:c3:dc:4e Broadcast ARP 60 Who has 192.168.0.242? Tell 192.168.0.1
363 15.037994054 48:9e:bd:9d:68:19 Broadcast ARP 60 Who has 192.168.0.184? Tell 192.168.0.1
367 15.297488761 ac:15:a2:b9:ce:ce Broadcast ARP 60 Who has 192.168.0.106? Tell 192.168.0.1
369 15.571380833 ac:15:a2:b9:ce:ce Broadcast ARP 60 Who has 192.168.0.114? Tell 192.168.0.1
Frame 12: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: HewlettP_cc:0d:43 (a0:8c:fd:cc:0d:43), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: HewlettP_cc:0d:43 (a0:8c:fd:cc:0d:43)
Sender IP address: 192.168.0.166
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.146
enp3s0: <live capture in progress>
Packets: 784 · Displayed: 186 (23.7%)
Profile: Default


Activities
Wireshark
Mon 15:11
(139) YouTube
https://www.youtube.com
*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
Expression...
No. Time Source Destination Protocol Length Info
73 4.196097234 192.168.0.243 23.212.50.234 TCP 66 50124 → 80 [ACK] Seq=1 Ack=1 Win=501 Le
84 5.179734680 192.168.0.243 192.168.0.204 TCP 54 7680 → 55724 [RST, ACK] Seq=1 Ack=1 Win
115 6.090416365 192.168.0.243 192.168.0.204 TCP 54 7680 → 55724 [RST, ACK] Seq=1 Ack=1 Win
144 6.206036032 192.168.0.243 192.168.0.204 TCP 54 7680 → 55724 [RST, ACK] Seq=1 Ack=1 Win
150 6.721403650 192.168.0.243 192.168.0.204 TCP 54 7680 → 55724 [RST, ACK] Seq=1 Ack=1 Win
Transmission Control Protocol, Src Port: 50124, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
Source Port: 50124
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window size value: 501
[Calculated window size: 501]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xc1ff [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
Transmission Control Protocol: Protocol
Packets: 9894 · Displayed: 3233 (32.7%)
Profile: Default
YouTube
Home
Shorts
Subscriptions
You
Your channel
History
Playlists
Your videos
Watch later
Liked videos
Your clips
Subscriptions
Republic World
Zee 24 Taas
NoCopyrightSou...
SHIROSUKI
Travelling Mantra
Call of Duty: Black Ops 6
FC Barcelona
GUILTY
INNOCENT
21:26
CAN I SURVIVE JUDGE SIMULATOR?
MM7Games
118K views • 18 hours ago
12:53


Activities
Wireshark
Mon 15:17
Thadomal Shahani Engin
https://tsec.edu
THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE
ABOUT | DEPARTMENTS | TRAINING & PLACEME
Admission 2024-25
Admission Notices 2024-25
Readmission 2024-25 (S.E., T.E., and B.E.)
Fees 2024-25
EXAMINATIONS / RESULTS
TRANSCRIPTS/DOCUMENTS
INTERNSHIPS
SCHOLARSHIP
MANDATORY DISPLAYS
NOTICES/ANNOUNCEMENTS
Academic Calendar
Upcoming Events
Sports Event
*enp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.srcport==443
Expression...
No. Time Source Destination Protocol Length Info
196 14.844935244 142.250.183.138 192.168.0.243 UDP 70 443 → 46901 Len=28
256 14.863342344 142.250.183.174 192.168.0.243 UDP 72 443 → 55534 Len=30
259 14.979186688 142.250.183.174 192.168.0.243 UDP 713 443 → 55534 Len=671
261 14.982056255 142.250.183.174 192.168.0.243 UDP 178 443 → 55534 Len=136
263 15.007065604 142.250.183.174 192.168.0.243 UDP 69 443 → 55534 Len=27
292 17.444728636 142.250.183.138 192.168.0.243 UDP 124 443 → 46901 Len=82
293 17.444949958 142.250.183.138 192.168.0.243 UDP 68 443 → 46901 Len=26
296 17.464993187 142.250.183.138 192.168.0.243 UDP 74 443 → 46901 Len=32
299 17.685583637 142.250.183.138 192.168.0.243 UDP 119 443 → 46901 Len=77
300 17.685662944 142.250.183.138 192.168.0.243 UDP 67 443 → 46901 Len=25
304 17.689616947 142.250.183.138 192.168.0.243 UDP 70 443 → 46901 Len=28
Frame 196: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
Ethernet II, Src: ac:15:a2:b9:ce:ce (ac:15:a2:b9:ce:ce), Dst: 04:0e:3c:1a:64:30 (04:0e:3c:1a:64:30)
Internet Protocol Version 4, Src: 142.250.183.138, Dst: 192.168.0.243
User Datagram Protocol, Src Port: 443, Dst Port: 46901
Source Port: 443
Destination Port: 46901
Length: 36
Checksum: 0xf315 [unverified]
[Checksum Status: Unverified]
[Stream index: 56]
Data (28 bytes)
0000 04 0e 3c 1a 64 30 ac 15 a2 b9 ce ce 08 00 45 00
0010 00 38 00 00 40 00 3b 11 37 95 8e fa b7 8a c0 a8
0020 00 f3 01 bb b7 35 00 24 f3 15 4a 4f d2 d9 a7 ad
0030 60 5b 68 bd 54 d5 ef 6e 85 9e e4 a4 5f b0 58 7b
0040 1b 45 b0 67 8b 2f


Activities
Wireshark
Thu 13:29
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC: $ tcpdump
Capturing from enp3s0 (tcp port http)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000000 192.168.0.243 34.107.221.82 TCP 66 35140 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=
2 0.000017274 192.168.0.243 34.107.221.82 TCP 66 35150 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=
Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: 04:0e:3c:1a:64:30 (04:0e:3c:1a:64:30), Dst: ac:15:a2:b9:ce:ce (ac:15:a2:b9:ce:ce)
Internet Protocol Version 4, Src: 192.168.0.243, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 35140, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
0000 ac 15 a2 b9 ce ce 04 0e 3c 1a 64 30 08 00 45 00
0010 00 34 ff 5d 40 00 40 06 7a 0d c0 a8 00 f3 22 6b
0020 dd 52 89 44 00 50 84 92 fd 34 d3 51 fc f0 80 10
0030 01 f5 00 d1 00 00 01 01 08 0a 45 0e a0 be 6b 38
0040 85 fa
enp3s0: <live capture in progress>
Packets: 8 · Displayed: 8 (100.0%)
Profile: Default



Conclusion:

Both **Wireshark** and **TCPDump** are valuable tools for network analysis, each suited to different scenarios. **Wireshark** is ideal for users needing in-depth analysis and a visual interface, while **TCPDump** is perfect for quick, efficient traffic captures in command-line environments. Together, they offer comprehensive network diagnostics capabilities for various use cases in cybersecurity, network monitoring, and troubleshooting.

LO3

Name: Saikarthik Iyer
Batch:T13
Roll no:41

Assignment No. 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.

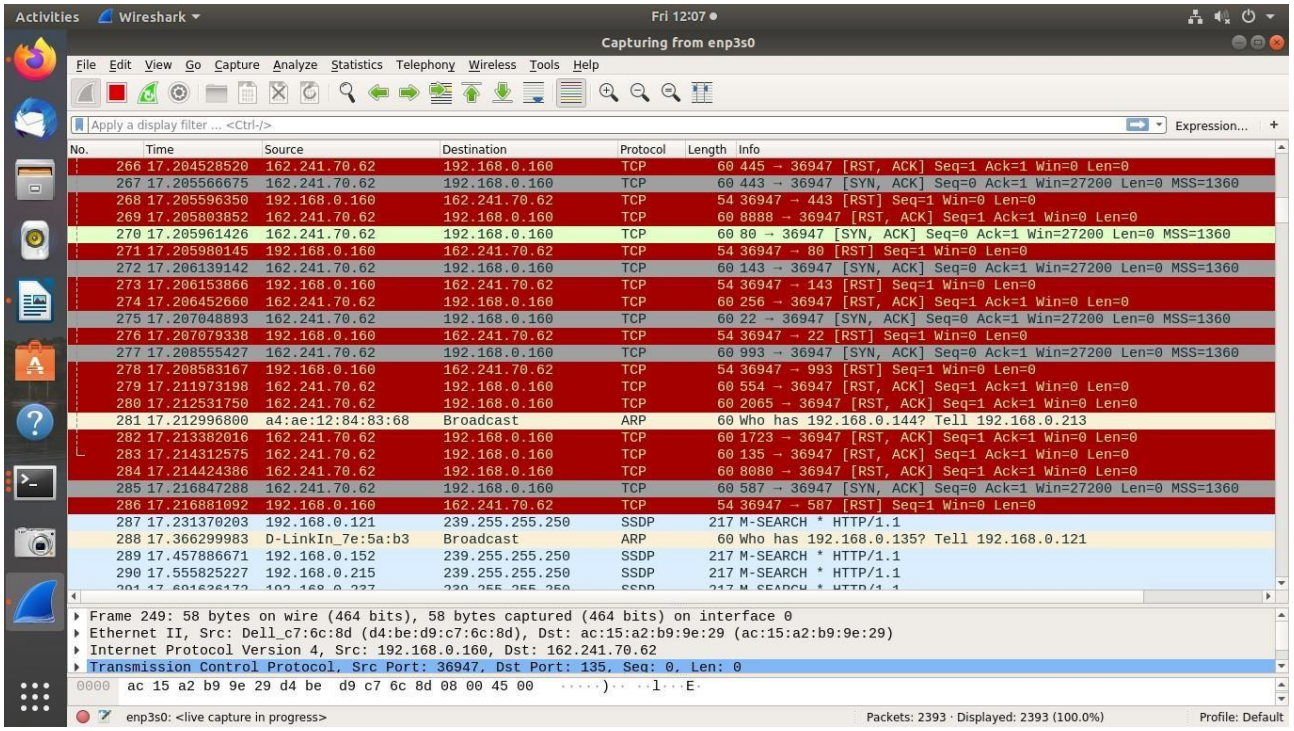
Theory:

1. **`-sS` (TCP SYN Scan)**:
   - A stealth scan that sends SYN packets to initiate the connection but doesn't complete the three-way TCP handshake. This scan is commonly used because it's faster and less likely to be logged by the target system.
2. **`-sT` (TCP Connect Scan)**:
   - Completes the full TCP handshake (SYN, SYN-ACK, ACK). It is used when the SYN scan is not available (e.g., non-root users). This scan is more likely to be logged by firewalls or intrusion detection systems.
3. **UDP Scan (`-sU`)**:
   - This scan identifies open UDP ports on the target system by sending UDP packets. Since UDP doesn't have handshakes like TCP, it can be slower and more difficult to interpret due to its stateless nature. 4. **`-sN` (Null Scan)**:
   - Sends packets with no TCP flags set. This scan is used to evade simple firewall rules, as no connection-related flags are present. 5. **`-sF` (FIN Scan)**:
   - Sends a packet with the FIN flag set. Some systems may treat this as an indication that a connection should be closed, and responses (or lack of them) can reveal information about open ports. 6. **`-sX` (Xmas Scan)**:
   - Sends packets with the FIN, PSH, and URG flags set. This "Xmas tree" scan can be used to detect open ports by analyzing how a system responds. 7. **`-sA` (ACK Scan)**:
   - Used primarily to map firewall rulesets, ACK scan helps determine whether ports are filtered or unfiltered by sending an ACK packet. 8. **`-sO` (IP Protocol Scan)**:
   - Scans for different IP protocols (like TCP, UDP, ICMP) on the target system, providing insights into which protocols are supported or active.
9. **`-p<port range>` (Port Range Selection)**:
   - Allows the user to specify a custom range of ports to be scanned. This option offers flexibility to target specific services or applications running on particular ports.
10. **`-F` (Fast Scan)**:
    - Scans only a limited number of well-known ports (about 100), which is faster than a full port scan. 11. **`-sv` (Service Version Detection)**:
    - Identifies the software version of services running on open ports. This helps in understanding what applications are present and whether they are vulnerable.
12. **`-O` (OS Detection)**:
    - Attempts to determine the operating system of the target based on its responses to Nmap's probes.
13. **`-v` (Verbose Mode)**:
    - Enables detailed output, which helps in analyzing the scanning process more thoroughly.

Output:

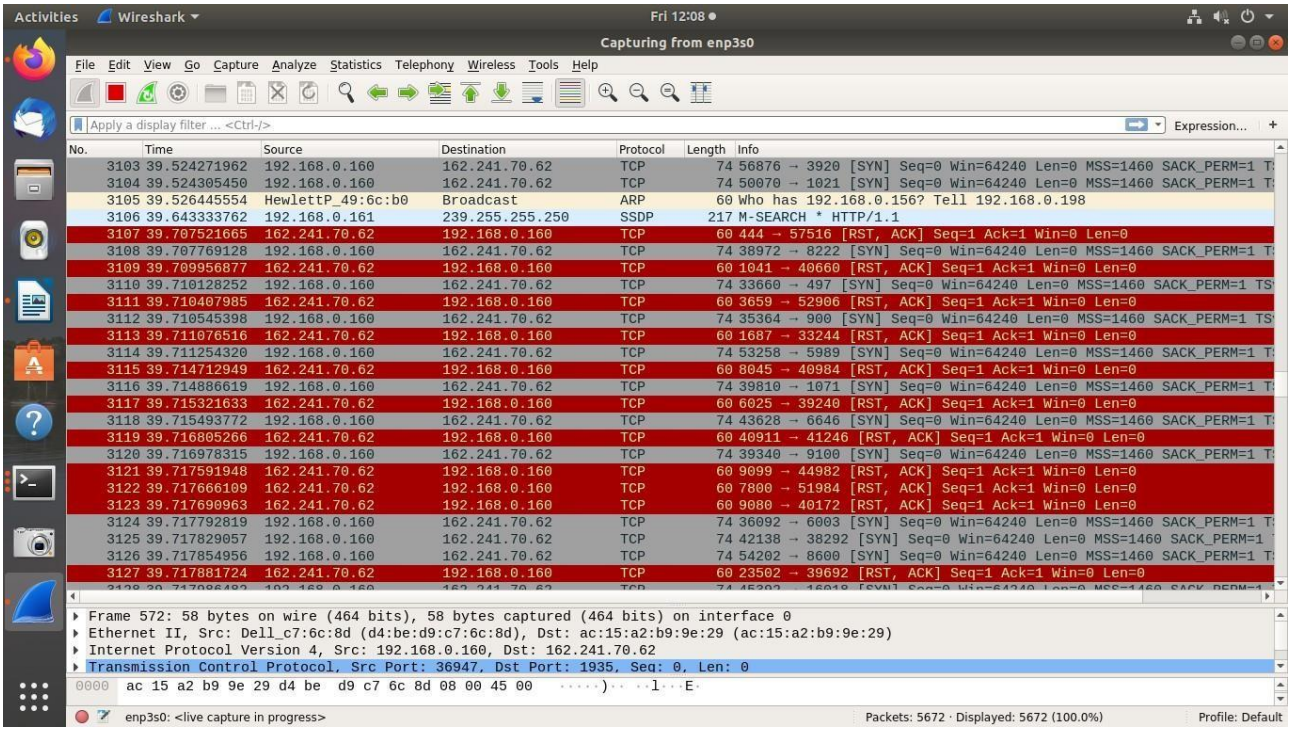
1.-sS

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sS tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:38 IST
Nmap scan report for tsec.edu (162.241.70.62)
Host is up (0.25s latency).
rDNS record for 162.241.70.62: 162-241-70-62.webhostbox.net
Not shown: 986 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
111/tcp  open     rpcbind
143/tcp  open     imap
161/tcp  filtered snmp
443/tcp  open     https
465/tcp  open     smtps
587/tcp  open     submission
993/tcp  open     imaps
995/tcp  open     pop3s
3306/tcp open     mysql

Nmap done: 1 IP address (1 host up) scanned in 4.64 seconds
lab902@lab902-OptiPlex-390:~$ S
```



-sT

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sT tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:40 IST
Nmap scan report for tsec.edu (162.241.70.62)
Host is up (0.23s latency).
rDNS record for 162.241.70.62: 162-241-70-62.webhostbox.net
Not shown: 986 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
111/tcp  open     rpcbind
143/tcp  open     imap
161/tcp  filtered snmp
443/tcp  open     https
465/tcp  open     smtps
587/tcp  open     submission
993/tcp  open     imaps
995/tcp  open     pop3s
3306/tcp open     mysql

Nmap done: 1 IP address (1 host up) scanned in 14.47 seconds
lab902@lab902-OptiPlex-390:~$
```



UDP Scan

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sU tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:43 IST
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 12.56% done; ETC: 11:49 (0:05:34 remaining)
Stats: 0:00:51 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.63% done; ETC: 11:50 (0:06:28 remaining)
Stats: 0:00:51 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.73% done; ETC: 11:50 (0:06:24 remaining)
Stats: 0:00:51 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.73% done; ETC: 11:50 (0:06:24 remaining)
Stats: 0:00:52 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.83% done; ETC: 11:50 (0:06:28 remaining)
Stats: 0:01:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 14.66% done; ETC: 11:54 (0:09:25 remaining)
Stats: 0:07:54 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 49.86% done; ETC: 11:58 (0:07:57 remaining)
Warning: 162.241.70.62 giving up on port because retransmission cap hit (10).
```

-sN

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sN tsec.org
[sudo] password for lab902:

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:46 IST
Nmap scan report for tsec.org (3.33.130.190)
Host is up (0.0024s latency).
Other addresses for tsec.org (not scanned): 15.197.148.33
rDNS record for 3.33.130.190: a2aa9ff50de748dbe.awsglobalaccelerator.com
All 1000 scanned ports on tsec.org (3.33.130.190) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 10.64 seconds
lab902@lab902-OptiPlex-390:~$
```

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sF tsec.org
[sudo] password for lab902:

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:51 IST
Nmap scan report for tsec.org (15.197.148.33)
Host is up (0.0025s latency).
Other addresses for tsec.org (not scanned): 3.33.130.190
rDNS record for 15.197.148.33: a2aa9ff50de748dbe.awsglobalaccelerator.com
All 1000 scanned ports on tsec.org (15.197.148.33) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 11.28 seconds
lab902@lab902-OptiPlex-390:~$
```

-sF

-sX

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sX tsec.org

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:54 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.11 seconds
lab902@lab902-OptiPlex-390:~$
```

-sA

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -sA tsec.org

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:56 IST
Nmap scan report for tsec.org (15.197.148.33)
Host is up (0.0026s latency).
Other addresses for tsec.org (not scanned): 3.33.130.190
rDNS record for 15.197.148.33: a2aa9ff50de748dbe.awsglobalaccelerator.com
All 1000 scanned ports on tsec.org (15.197.148.33) are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.39 seconds
```

-sO

```
All 1000 scanned ports on tsec.org (15.197.148.33) are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.39 seconds
lab902@lab902-OptiPlex-390:~$ sudo nmap -sO tsec.org

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 11:56 IST
Nmap scan report for tsec.org (15.197.148.33)
Host is up (0.0021s latency).
Other addresses for tsec.org (not scanned): 3.33.130.190
rDNS record for 15.197.148.33: a2aa9ff50de748dbe.awsglobalaccelerator.com
All 256 scanned ports on tsec.org (15.197.148.33) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 6.61 seconds
lab902@lab902-OptiPlex-390:~$
```

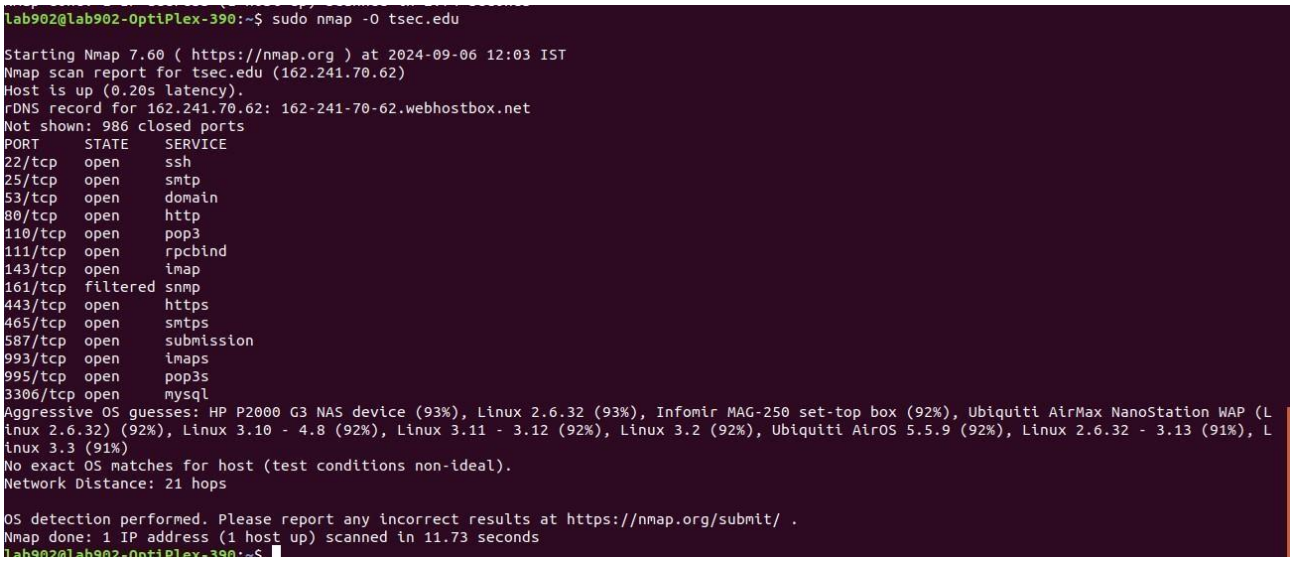
-p<port range>

```
lab902@lab902-OptiPlex-390:~$ nmap -p 1-100 tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 12:01 IST
Nmap scan report for tsec.edu (162.241.70.62)
Host is up (0.22s latency).
rDNS record for 162.241.70.62: 162-241-70-62.webhostbox.net
Not shown: 96 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
25/tcp open  smtp
53/tcp open  domain
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 2.36 seconds
```

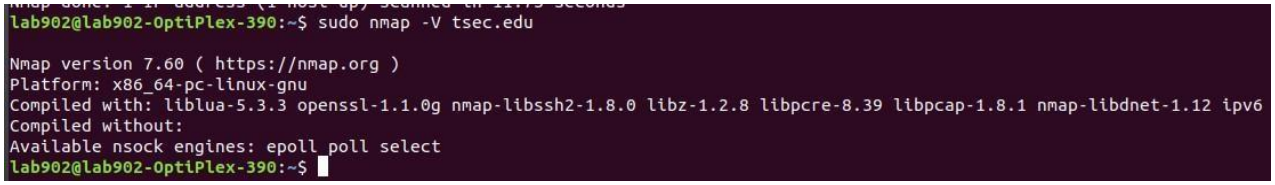
-F

```
lab902@lab902-OptiPlex-390:~$ sudo nmap -F tsec.edu

Starting Nmap 7.60 ( https://nmap.org ) at 2024-09-06 12:03 IST
Nmap scan report for tsec.edu (162.241.70.62)
Host is up (0.23s latency).
rDNS record for 162.241.70.62: 162-241-70-62.webhostbox.net
Not shown: 87 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
111/tcp  open  rpcbind
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 2.74 seconds
lab902@lab902-OptiPlex-390:~$ sudo nmap -O tsec.edu
```

-0



-v



Conclusion:

Nmap is a powerful and versatile tool for network exploration and security auditing. By using different scanning techniques, Nmap can provide valuable insights into the target's network, services, and security posture. Each scanning option is designed to bypass or avoid detection from firewalls and intrusion detection systems in different ways, offering security professionals multiple methods for gathering information. The choice of scan depends on the type of data you need (open ports, operating system, etc.) and the security setup of the target system.

LO4

Saikarthik Iyer
T1341

Aim : To simulate DOS attack using Hping3.

Theory :

Understanding Denial of Service (DoS) Attack

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning and availability of a network, system, service, or resource. The primary goal of a DoS attack is to overwhelm the target with a flood of traffic or to exploit vulnerabilities in such a way that it becomes inaccessible to its legitimate users. Let's delve deeper into some common types of DoS attacks:

SYN Flood Attack

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

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

ICMP Flood Attack (Ping Flood Attack)

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

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

Saikarthik Iyer
T1341

SMURF Attack

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

Here's how a SMURF attack works:

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

Hping3 Commands for SYN Flood and ICMP Flood

SYN Flood using Hping3

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

- `-c 15000`: Specifies sending 15,000 packets.
- `-d 120`: Sets the data size to 120 bytes in each packet.
- `-S`: Sets the SYN flag in TCP packets.
- `-w 64`: Defines a window size of 64.

Saikarthik Iyer
T1341

- `-p 80`: Targets port 80 (commonly used for HTTP).

- `--flood`: Floods the target with packets continuously.

- `--rand-source`: Utilizes random source IP addresses.

- `192.168.1.159`: Specifies the target IP address.

ICMP Flood using Hping3

```bash hping3 -1 --flood -a 192.168.103 192.168.1.255

```

- `-1`: Indicates the use of ICMP echo (ping) requests.

- `--flood`: Initiates the continuous flooding of the target.

- `-a 192.168.103`: Spoofs the source IP address as 192.168.103.

- `192.168.1.255`: Targets the broadcast address, causing multiple devices on the network to

  respond.

Example Hping3 Command for a SMURF Attack

In a SMURF attack, Hping3 can be used to generate ICMP echo-request packets and send them to the broadcast address of an intermediate network, causing amplification and flooding. Below is an example command:

```bash hping3 -1 --flood -a <spoofed_source_ip> <broadcast_address>

```

- `-1`: Indicates the use of ICMP echo (ping) requests.

- `--flood`: Initiates the continuous flooding of the target.

- `-a <spoofed_source_ip>`: Spoofs the source IP address as `<spoofed_source_ip>`. The

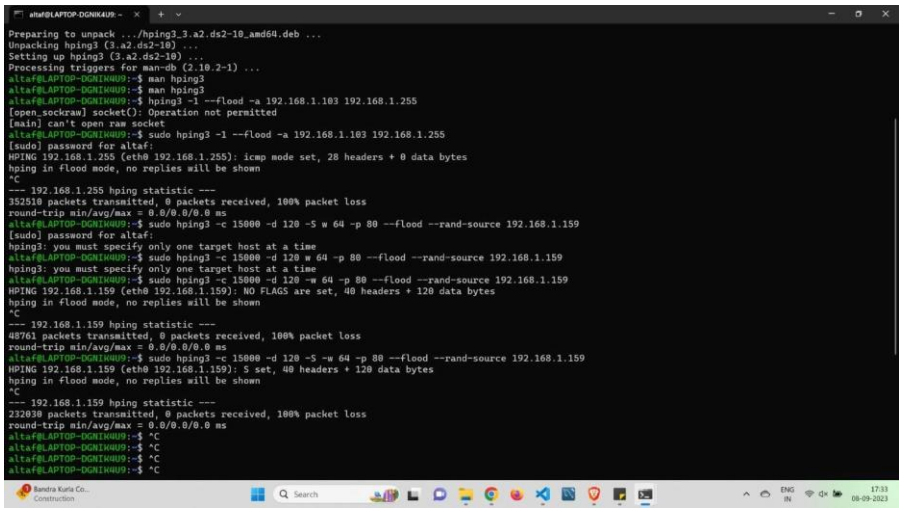
  attacker often uses a spoofed IP address to hide their identity.

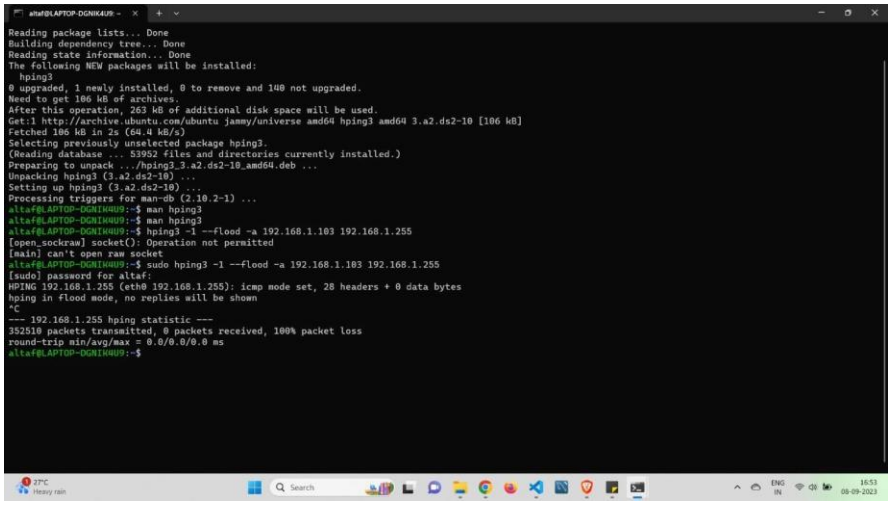
- `<broadcast_address>`: Specifies the broadcast address of the intermediate network. This is

  where the ICMP requests are sent.

Saikarthik Iyer
T1341

Output:





Conclusion: Used open-source tools to scan the network for vulnerabilities and simulate attacks(LO4 is achieved).

Saikarthik Iyer
T1341

Aim : To study and configure firewalls using IP table.

Theory :

Firewall:

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

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

Types of Firewalls

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

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

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

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

Basic of iptables:

Iptables is a firewall, installed by default on all official Ubuntu distributions (Ubuntu, Kubuntu, Xubuntu). When you install Ubuntu, iptables is there, but it allows all traffic by default.

Saikarthik Iyer
T1341

The rules in IPTables are written to deal 3 different scenarios:

1.Those packets entering your machine that are destined for your machine. (INPUT) 2.Those packets leaving your machine. (OUTPUT)

3.Those packets entering your machine, but are destined for another machine and will pass through your machine (FORWARD).

In Iptables, these scenarios are referred to as INPUT, OUTPUT, and FORWARD, respectively.

Once the traffic type has been specified, three actions may be taken:

1.ACCEPT allows packets to pass through the firewall.

2.DROP ignores the packet and sends no response to the request.

3.REJECT ignores the packet, but responds to the request with a packet denied message.

Output:

Saikarthik Iyer
T1341

```
shawn@Shawn-Laptop:~$ sudo iptables -L
[sudo] password for shawn:
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
```

```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
shawn@Shawn-Laptop:~$ sudo 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
```

```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

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

```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

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

Saikarthik Iyer
T1341

```
shawn@Shawn-Laptop:~$ sudo iptables -I INPUT 1 -i lo -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

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

```
shawn@Shawn-Laptop:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     any     anywhere             anywhere
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    3   486 DROP       all  --  any    any     anywhere             anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p icmp -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

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

```
shawn@Shawn-Laptop:~$ sudo iptables -F
shawn@Shawn-Laptop:~$ sudo 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
```

Saikarthik Iyer
T1341

```
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
52 packets transmitted, 0 received, 100% packet loss, time 53007ms

shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p icmp -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7288ms
```

```
shawn@Shawn-Laptop:~$ sudo iptables -A OUTPUT -p icmp -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -
Bad argument `-'
Try `iptables -h' or 'iptables --help' for more information.
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
36 packets transmitted, 0 received, 100% packet loss, time 36405ms
```

Saikarthik Iyer
T1341

Types of iptables:

I. IPTABLES TABLES and CHAINS

IPTables has the following 4 built-in tables.

1. Filter Table

Filter is default table for iptables. So, if you don't define you own table, you'll be using filter table. Iptables's filter table has the following built-in chains.

- INPUT chain – Incoming to firewall. For packets coming to the local server.
- OUTPUT chain – Outgoing from firewall. For packets generated locally and going out of the local server.
- FORWARD chain – Packet for another NIC on the local server. For packets routed through the local server.

Type the following command and see the result sudo iptables -t filter -L

2. NAT table

Iptable's NAT table has the following built-in chains.

- PREROUTING chain – Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing). This helps to translate the destination ip address of the packets to something that matches the routing on the local server. This is used for DNAT (destination NAT).
- POSTROUTING chain – Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the desintation server.

This is used for SNAT (source NAT).

- OUTPUT chain – NAT for locally generated packets on the firewall. Type the following command and see the result sudo iptables -t nat -L

3. Mangle table

Saikarthik Iyer
T1341

Iptables's Mangle table is for specialized packet alteration. This alters QOS bits in the TCP header. Mangle table has the following built-in chains.

• PREROUTING chain

• OUTPUT chain

• FORWARD chain

• INPUT chain

• POSTROUTING chain

Type the following command and see the result

sudo iptables -t nat -L

4. Raw table

Iptable's Raw table is for configuration excemptions. Raw table has the following built-in chains.

• PREROUTING chain

• OUTPUT chain

```
shawn@Shawn-Laptop:~$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere
```

Saikarthik Iyer
T1341

```
shawn@Shawn-Laptop:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

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

```
shawn@Shawn-Laptop:~$ sudo iptables -t mangle -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

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

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

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

Saikarthik Iyer
T1341

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

Theory :

1. Installing Snort

- **Installation**: Snort is available for both Linux and Windows. The installation involves downloading the Snort package from its official source and following the setup process. During installation, you specify the network interface that Snort will monitor.

2. Adding Rules

- **Rules**: Snort uses predefined rules to detect specific types of network activity that could indicate malicious behavior. These rules define patterns, actions to take (such as logging or alerting), and the traffic to inspect. Users can create custom rules or use community-contributed rule sets.
- **Structure**: A Snort rule consists of an action (alert, log, etc.), protocol, source/destination IP addresses, ports, and specific options that define the detection logic.

3. Configuring Snort

- **Configuration File**: The main Snort configuration file specifies the network variables, rule paths, and preprocessors (used for advanced traffic detection). It also defines how Snort handles and logs alerts and what traffic patterns to monitor (such as internal vs. external networks).
- **Preprocessors**: These are modular add-ons that extend Snort's capabilities, enabling it to detect various network anomalies, such as port scanning or fragmented packets.

4. Validating Configuration

- **Validation**: Before running Snort, it is important to validate the configuration to ensure that there are no syntax errors or misconfigurations. This process checks the integrity of the configuration file and ensures all rules and preprocessors are correctly set up.

Saikarthik Iyer
T1341

5. Monitoring for Intrusions

- **Running Snort in IDS Mode**: Once Snort is configured, it can be run in intrusion detection mode. In this mode, Snort monitors network traffic in real-time and checks for matches against the active rule sets. When malicious traffic is detected, Snort generates alerts.
- **Alerting and Logging**: Snort can be configured to log alerts in various formats, such as text files or centralized logging systems. Alerts can be displayed on the console or sent to external logging services for further analysis.

6. Monitoring and Analyzing Logs

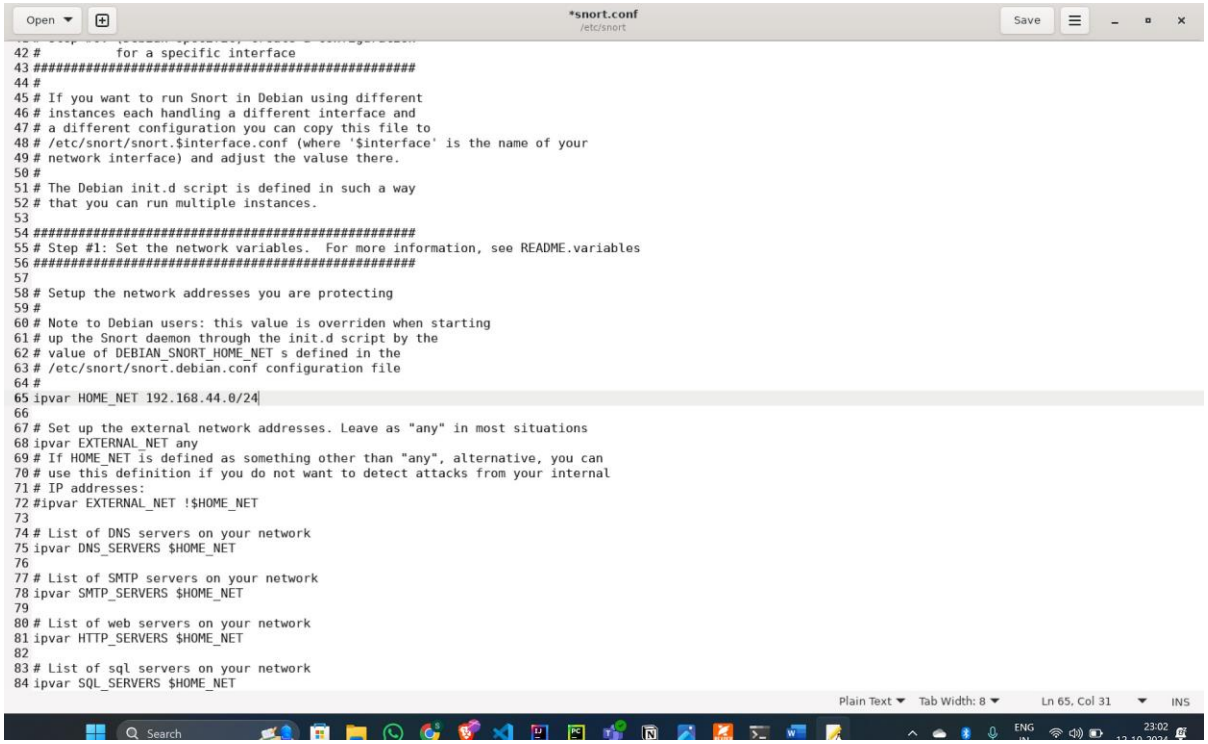
- **Log Review**: Regular log monitoring is crucial for intrusion detection. Administrators can analyze logs manually or use web-based interfaces to visualize and manage alerts more effectively.
- **Integration with Tools**: For more efficient monitoring, Snort can be integrated with visualization and reporting tools like Snorby or BASE, which provide a graphical interface for analyzing intrusion alerts and trends over time.

This process provides a robust way to detect and respond to network-based attacks using Snort IDS.

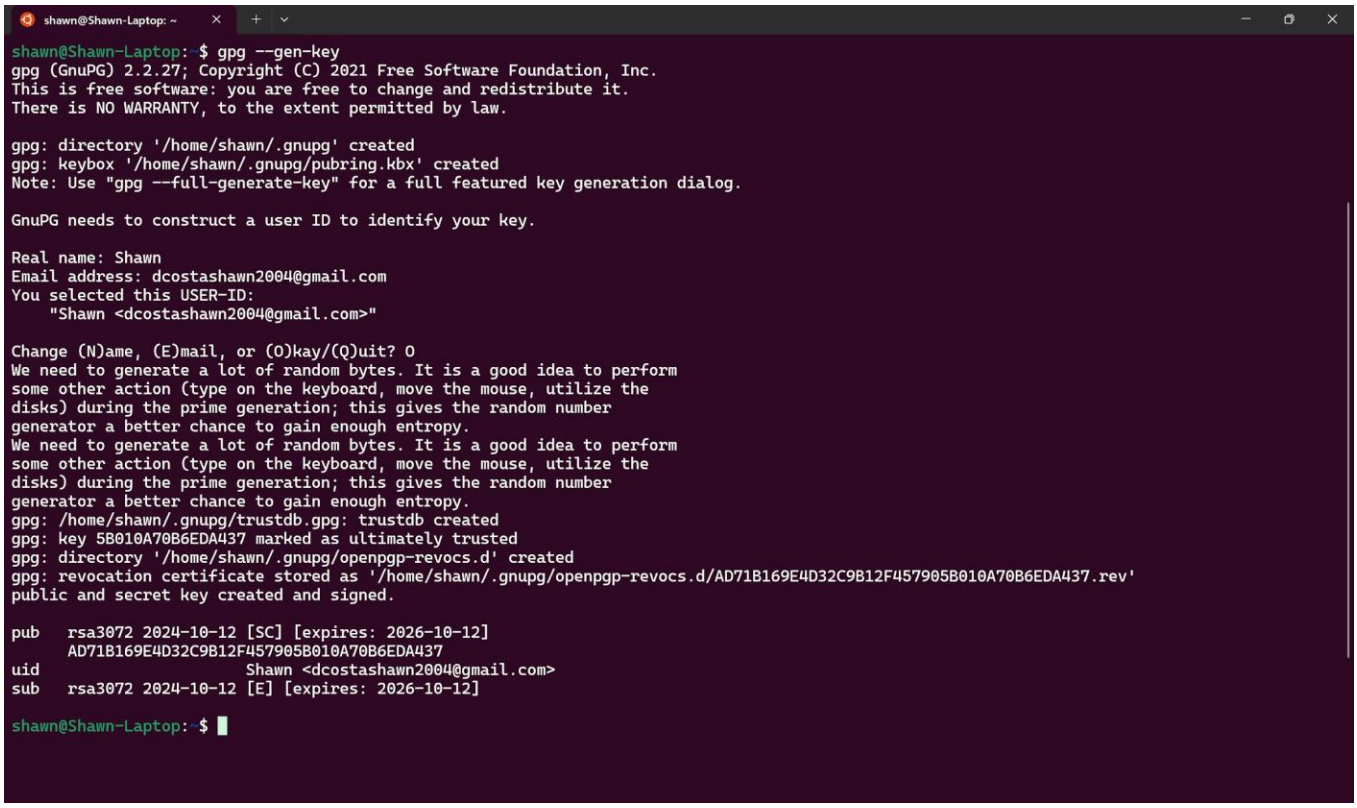
Output:

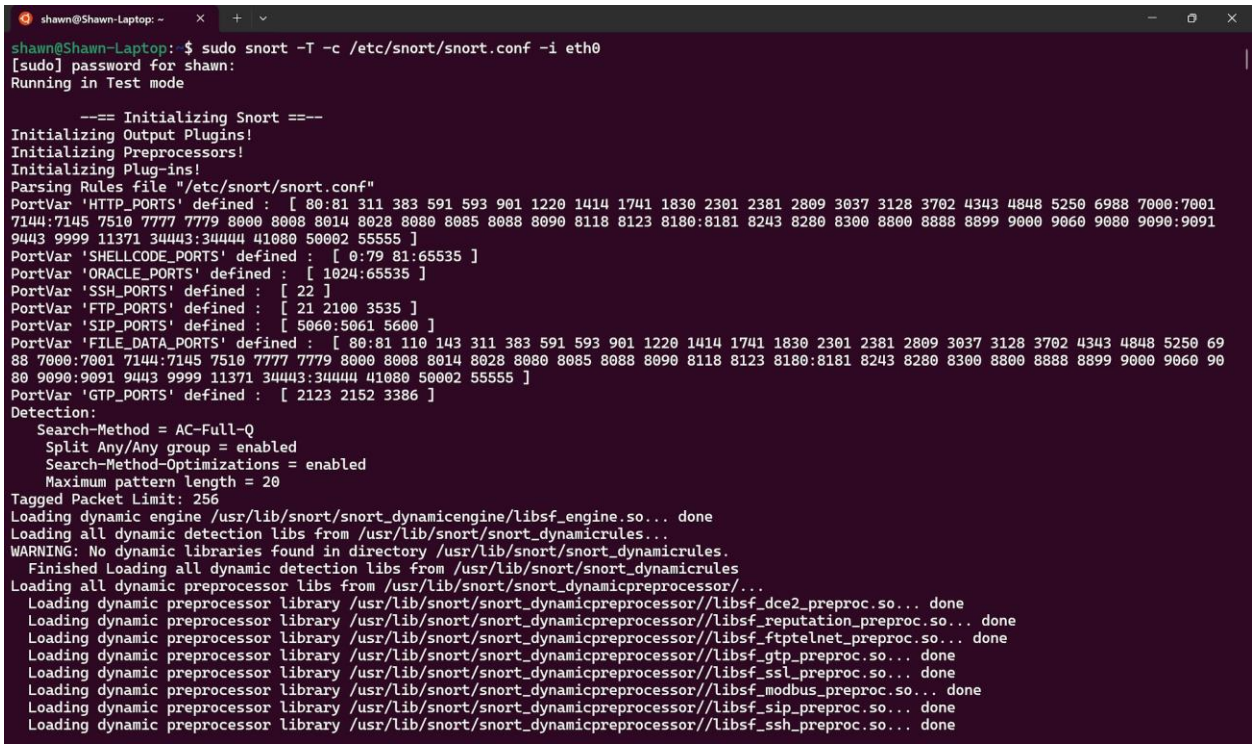
sudo gedit /etc/snort/snort.conf

Saikarthik Iyer
T1341

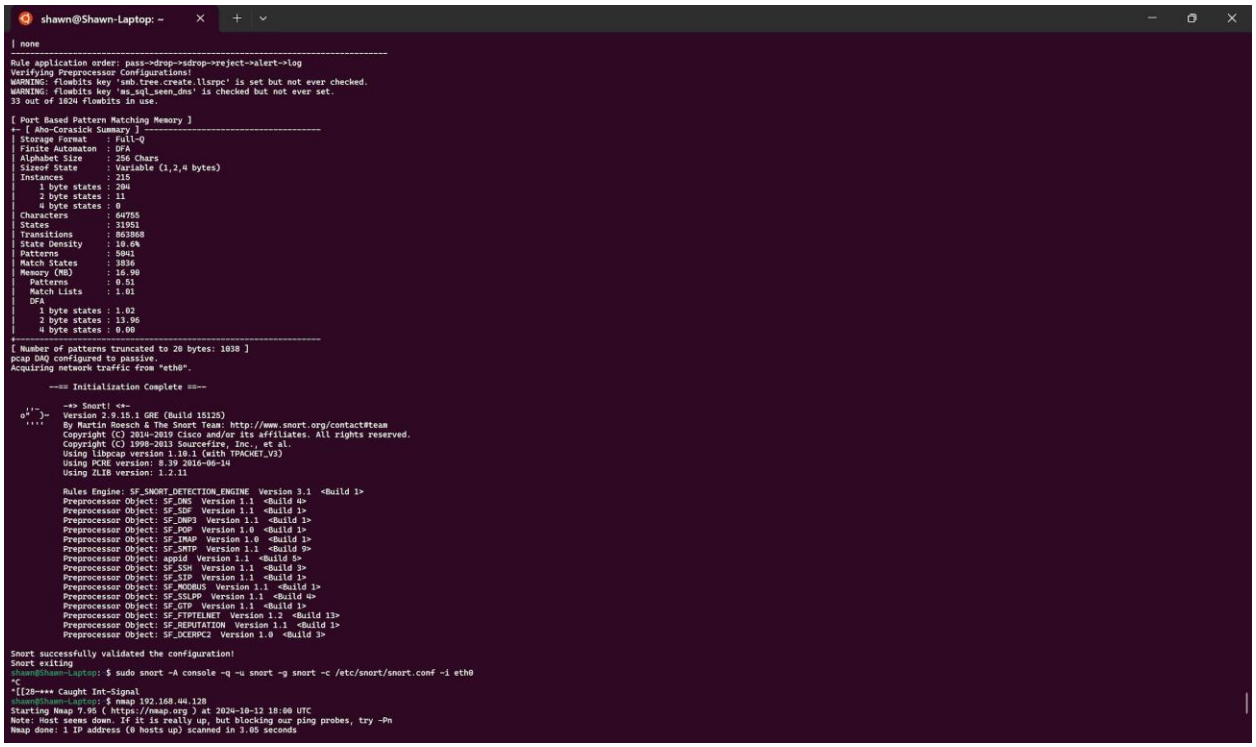


Saikarthik Iyer
T1341





Saikarthik Iyer
T1341



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

Saikarthik Iyer
T1341

Aim : To explore the GPG tool of linux to implement email security.

Theory :

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

PGP using GPG:

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

2. **Data Encryption**:

   - **Symmetric Encryption**: For encrypting the actual message, a random symmetric key (session key) is generated. The message is encrypted with this key using a symmetric algorithm like AES.

   - **Asymmetric Encryption**: The session key itself is then encrypted with the recipient's public key using asymmetric encryption (like RSA). This allows the session key to be securely shared.

3. **Digital Signature**:

   - The sender signs the message with their private key, which helps ensure **authenticity** and **integrity**. This digital signature proves the identity of the sender and verifies that the message hasn’t been tampered with.

   - The recipient can verify the signature using the sender's public key.

4. **Key Management**:

   - GPG uses **keyrings** to manage the public and private keys. Users can add trusted public keys and use them to encrypt data.

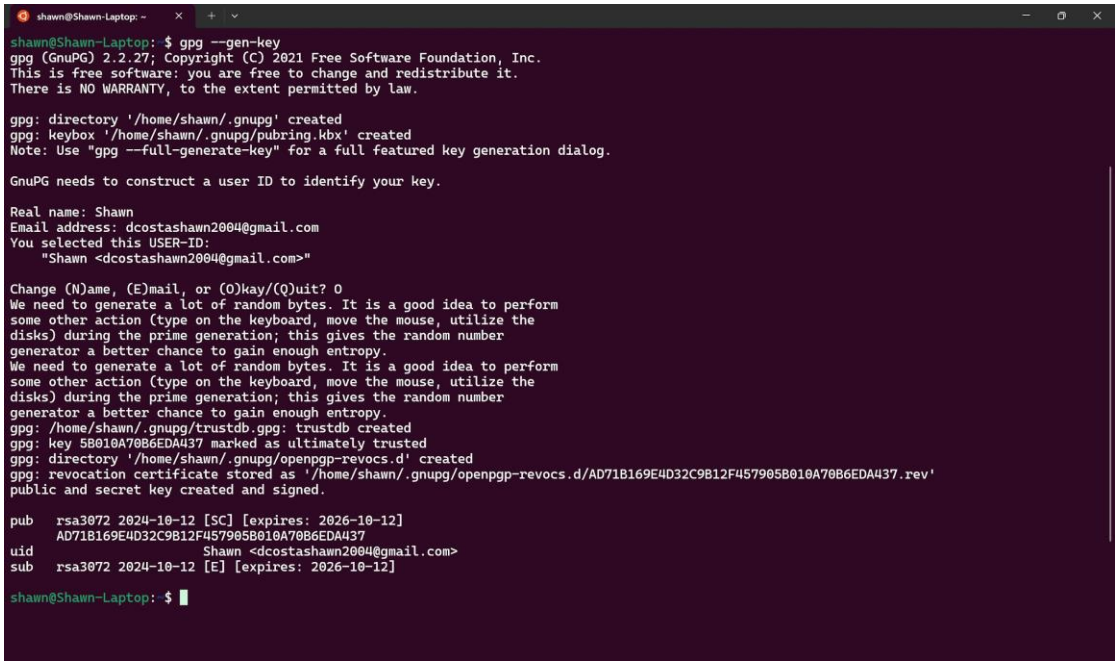
   - GPG also supports **key servers**, where public keys can be shared and retrieved.

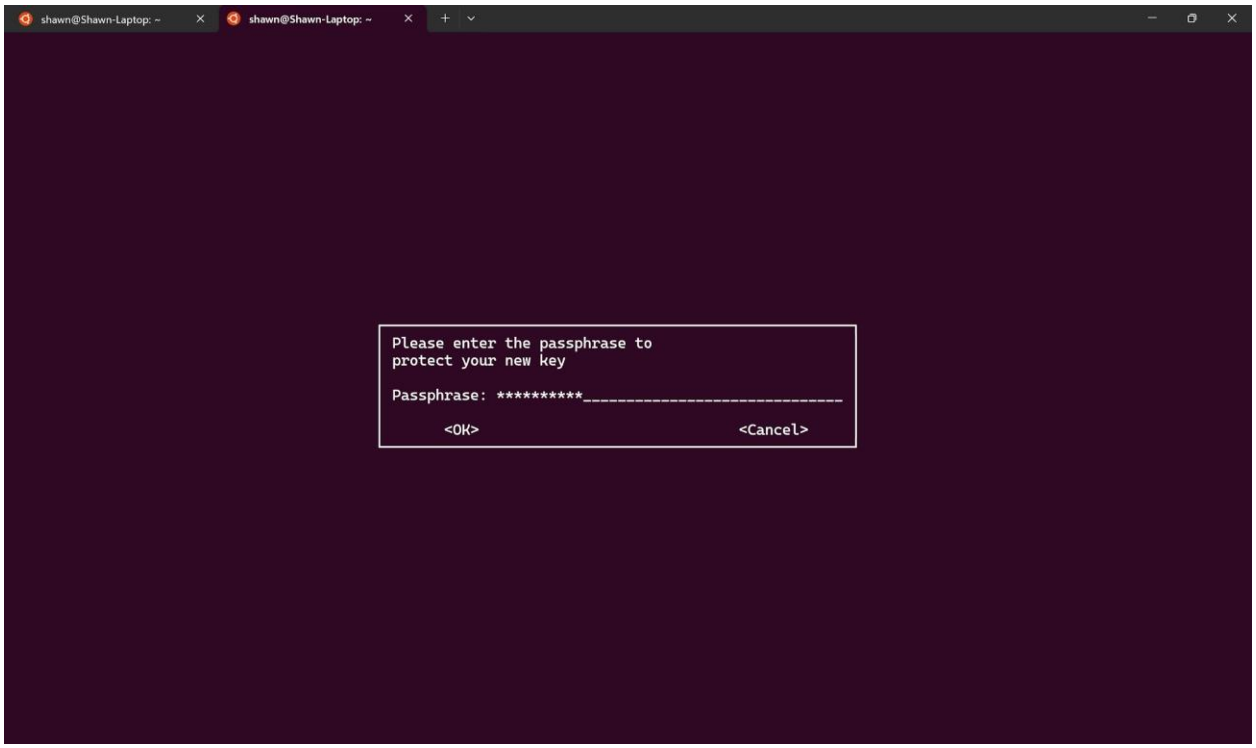
Saikarthik Iyer
T1341

5. **Web of Trust**:
    - Unlike centralized systems like SSL, PGP uses a **web of trust** for identity verification. Users sign each other's public keys, building a network of trust without relying on a single central authority.
6. **Message Decryption**:
    - When the recipient receives the message, they use their private key to decrypt the symmetric session key. Once they have the session key, they can decrypt the message.
7. **Passphrase Protection**:
    - To enhance security, private keys are often encrypted with a passphrase, adding an extra layer of protection in case the private key is compromised.

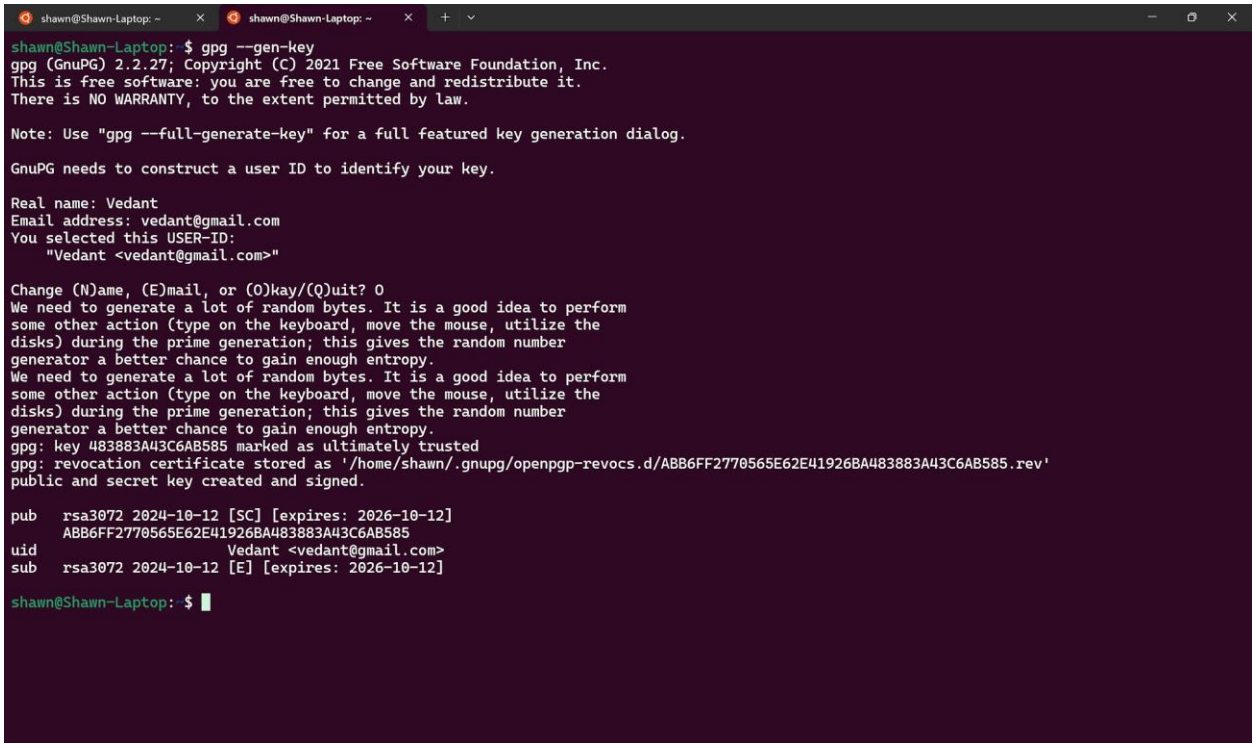
Output:

Saikarthik Iyer
T1341

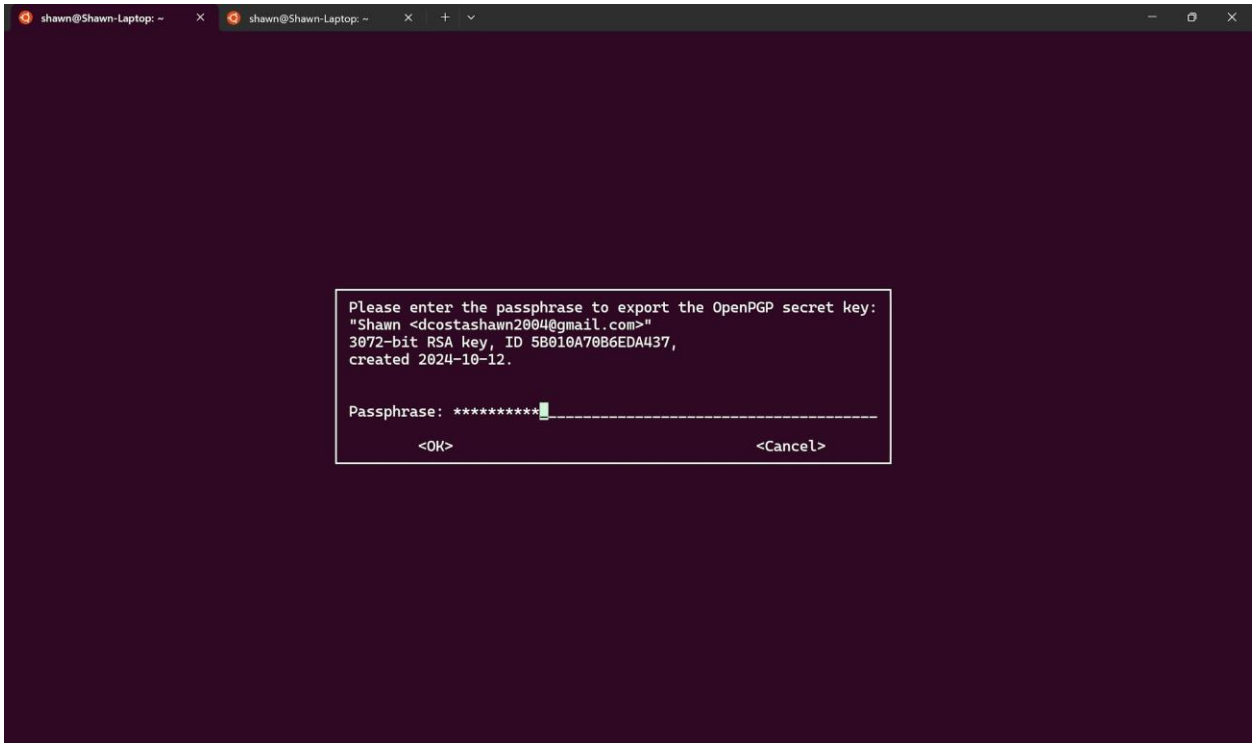




Saikarthik Iyer
T1341



Saikarthik Iyer
T1341




```
shawn@Shawn-Laptop:~$ gpg --fingerprint vedant@gmail.com
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2026-10-12
pub   rsa3072 2024-10-12 [SC] [expires: 2026-10-12]
      ABB6 FF27 7056 5E62 E419  26BA 4838 83A4 3C6A B585
uid           [ultimate] Vedant <vedant@gmail.com>
sub   rsa3072 2024-10-12 [E] [expires: 2026-10-12]
```

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

Saikarthik Iyer
T1341

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

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

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

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


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

     Vedant <vedant@gmail.com>

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

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

```
shawn@Shawn-Laptop:~$ vi test.txt
shawn@Shawn-Laptop:~$ gpg --encrypt -r vedant@gmail.com test.txt
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2026-10-12
```

Saikarthik Iyer
T1341

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

Saikarthik Iyer.
T13 41

Written Assignment - 1

1. Explain Instruction detection System (Definition, Classification, Advantage, Disadvantages and applications)

→ Definition

An Instruction detection System (IDS) is a security tool designed to detect malicious behaviour or anomalies at the instruction execution level within a computer system or a specific application. The system continuously monitors the execution level within a computer machine level instructions or code paths to identify any deviations from expected behaviour patterns. This is particularly useful in systems where traditional security measures like firewalls or antivirus software may not be sufficient, such as embedded systems or IOT devices.

→ Classification of Intrution detection Systems, IDS can be classified into various categories based on different criteria. The two most prominent classifications are:-

i Based on Detection approach

→ Signature based detection. This

type relies on a predefined - based detection. This type relies on a predefined set of instruction or patterns or signatures that are known to be malicious. The system detects threats by comparing real time intrusions against these signatures.

i. Based on coverage

⇒ Host - based Instruction detection system (HIDS), monitors instruction activities within a particular host, such as a PC, mobile device, or server. The focus is on monitoring local instructions executed by applications or the OS.

⇒ Network based Instruction detection system (NIDS): In the context of distributed systems, NIDS monitors instruction flows and activities across networked devices.

⇒ Advantages of Instruction detection systems.

i. Deep - level monitoring: IDS operates at the instruction

level, allowing it to detect threats or anomalous behaviour that might be invisible to traditional security systems.

ii) Detection of low-level threats: Because it focuses on instruction execution, IDS can detect sophisticated attacks like buffer overflows, zero-day exploits and rootkits.

iii) Real time detection - By monitoring instructions in real-time, one IDS can provide immediate alert and responses to threats minizing damage.

iv) Enhanced security for critical systems. It's particularly useful in securing embedded systems, IOT devices, and industrial control systems where traditional security methods may not be sufficient.

⇒ Disadvantages of Instruction Detection Systems

i) High complexity: Monitoring and analyzing instructions in real-time requires significant computational power and can be

complex to implement and manage, especially in large systems.

ii Performance overhead: Instruction detection systems can introduce a noticeable performance penalty on the system they are protecting due to the constant monitoring of every instruction executed

iii False positives: Anomaly-based
⇒ IDS may generate a large number of false positives by flagging legitimate but unusual activities as threats, leading to unnecessary alarms and disruptions

iv Limited in scope (Signature-based)
⇒ Signature based IDS can only detect known threats and might miss new or modified malware that doesn't match the pre existing instruction patterns

⇒ Applications of Instruction Detection Systems:

i Malware detection: IDS is commonly used to detect and prevent malware attacks by identifying suspicious instruction sequences that indicate malicious behaviour

ii) Embedded Systems Security: In mission-critical environments, such as medical devices or industrial control systems, IDS helps to ensure that instructions executed by the process or follow expected patterns, safeguarding against cyberattacks.

iii) IOT devices: IOT devices often lack robust security measures and IDS can monitor the instructions processed by these devices to detect and prevent unauthorized activities

iv) Critical Infrastructure Protection: Instruction detection systems are employed in securing the computing infrastructure that controls critical operations such as power grids, water treatment plants and transportation systems.

2 What is DOS and DDOS attack? Explain different types of these attacks

⇒ DOS (Denial of Service) and DDOS (distributed denial of service) are two types of cyberattacks that aim to disrupt the normal functioning of targeted server, network or

service by overwhelming it with a flood of illegitimate requests, both attacks prevent legitimate users from accessing the targeted resource but they differ in their methods of execution. DoS attack - In a DoS attack, a single machine used to flood the target system with overwhelming traffic, causing it to slow down or crash.

- DDOS attack : A DDoS attack is more sophisticated, involving multiple systems & often compromised machines (forming a botnet) to send traffic to the target. This makes DDoS attacks more difficult to defend against as the traffic appears to come from various locations, making it harder to block.

3 Types of DoS and DDoS attacks.

⇒ Volume - based Attacks.

These attacks are aimed at overwhelming the bandwidth or capacity of a targeted network or system. The main goal is to flood the network with massive amounts of data or traffic, rendering it unusable.

- UDP flood (User Datagram Protocol Flood): In a UDP flood, the attacker sends large numbers of UDP packets to random ports on the target machine. The target system attempts to process the packets and respond with ICMP
  - ⇒ DoS: Single-source UDP flood
  - ⇒ DDoS Multiple systems flood the target with UDP packets
- ICMP floods (ping flood): This attack sends a large number of ICMP Echo Request (ping) packets to a target. The target system consumes its resources in responding to each ping request.
  - > DoS: Smurf source sends ping requests.
  - > DDoS: Botnet or distributed machines send ICMP requests
- HTTP flood: This attack simulates legitimate-looking requests by flooding the target's Web Server with HTTP GET or POST requests
  - > DoS: one machine
  - > DDoS: Many devices send HTTP requests, simulating multiple users.

- SYN flood : A SYN flood targets the TCP three-way handshake process. the attackers sends numerous SYN requests to the target system, which allocates resources to establish connections. ~~However,~~
  - > DoS : Single machine sends SYN requests without completing the handshake.
  - > DDoS : Multiple devices initiate incomplete TCP handshakes.

>> Protocol-based attacks:
Protocol attacks exploit weakness in the network layer or protocol implementation to overwhelm the target's system resources like CPU or memory.

- Ping of death : This attack sends malformed or oversized ICMP packets to a target. The target cannot handle these packets and crashes or becomes unstable.
  - > DoS - Single-source sends malicious pings
  - > DDoS : Multiple sources sends oversized pings.
- Smurf attack : In a smurf attack, the attacker sends ICMP echo request (ping) packets with a spoofed source

IP (the target's IP) to a network's broadcast address.

- DoS: Single machine spoofs the target's IP.
- DDoS: Multiple machines send spoofed packets to amplify the attack.

- Fraggle attack: Similar to a Smurf attack, but instead of using ICMP packets, a Fraggle attack uses UDP packets to a network's broadcast address, causing the same overwhelming response.
- DoS: Single-source sends UDP packet to a broadcast address.
- DDoS: Multiple machines participate in the attack using UDP packets.

- DNS Amplification: It is a reflection based attack where the attacker sends small DNS queries to open DNS resolvers, with the source IP address spoofed to the target's IP.
- DoS: Single-source sends spoofed queries
- DDoS: Multiple machines use DNS servers to amplify traffic towards the target

32. Application - layer Attacks.
These attacks target specified applications, like web servers or databases, with the goal of exhausting their resources (eg. memory or disk space).
- Slowloris - Slowloris is a low-bandwidth attack that (sends)

22/10/24

Saikarthik Iyer
T1341

SL Written Assignment - 2

Q1 Explain the working of elliptic curve digital signature and its benefits over RSA digital signature.

→ Elliptic curve digital signature Algorithm is a variant of the digital signature algorithm that uses elliptic curve cryptography (ECC) to provide secure digital signatures

- Working

1) Key generation:

→ Choose a random integer d as the private key, where d is in the range $1 \leq d \leq n-1$ and n is the order of the elliptic curve.

→ Compute the public $Q = d \cdot G$, where 'G' is a fixed point on the elliptic curve known as the generator point and 'd' is the private key.

2) Signing process:

→ To sign a message 'm', the signer hashes the message using a cryptographic hash function $H(m)$.

⇒ Choose a random integer 'k' in the range $1 \leq k \leq n-1$

⇒ Computes, the point $R = k.G$ on the elliptic curve, where $R = (x_1, y_1)$

⇒ The signature is composed of two values r and s:

$r = x_1 \bmod n$

$s = k^{-1}(H(m) + r.d) \bmod n$

⇒ The signature is then the pair (r, s)

3 Verification Process

⇒ To verify a signature (r, s), the verifier

⇒ Compute two values:

$u_1 = H(m).s^{-1} \bmod$

$u_2 = rs^{-1} \bmod n$

⇒ The signature is valid if $r = x_1 \bmod n$, where $R = (x_1, y_1)$

∴ Benefits of ECDSA over RSA digital signature

⇒ Smaller key sizes

⇒ ECDSA offers equivalent security to RSA with much smaller key sizes

⇒ This leads to faster computations and less storage and

transmission requirements.

2 Higher Efficiency

⇒ The smaller key size results in faster key generation signing and verification in ECDSA compared to RSA.

⇒ This efficiency is especially beneficial for devices with limited resource such as smartphones or IoT devices.

3 Lower Bandwidth and storage requirements

⇒ Because ECDSA uses smaller keys and signature than RSA, the digital signatures takes up less space.

⇒ This translates to reduced bandwidth and storage, which is useful for network communications and file systems.

4 Stronger security per bit

⇒ For a given key size, ECDSA provides a higher level of security than RSA.

classmate
Date
Page

→ This makes it more future-proof in terms of resistance to potential advances in cryptanalysis or computing power.

22/10/24