Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that ~~Mr.~~/Miss SHREYA SANDEEP KAMATH

of IT Department, Semester V with

Roll No. 53 has completed a course of the necessary

experiments in the subject SECURITY LAB under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

20/10/23

Teacher In-Charge

Head of the Department

Date _____

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

THADOMAL SHAHANI TSEC ENGINEERING SOLLEGE

CONTENTS

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

LAB ASSIGNMENT NO. 1

AIM: Breaking Shift Cipher and Mono-alphabetic Substitution cipher using frequency analysis method.

LAB OUTCOME ATTAINED:

LO1: Illustrate symmetric cryptography by implementing classical ciphers.

THEORY:

SHIFT CIPHER

A shift cipher, also known as the Caesar cipher, is one of the simplest and oldest forms of encryption techniques. It is a substitution cipher where each letter in the plaintext is shifted a certain number of positions down the alphabet. This number is called the "key" or "shift value."

For example, with a shift value of 3, the letter "A" would be encrypted to "D," "B" to "E," and so on. The process wraps around the alphabet, so "X" would be encrypted to "A," "Y" to "B," and "Z" to "C."

The Caesar cipher can be broken using a brute-force attack because it only has 25 possible keys (shift values). With a limited number of options, an attacker can quickly try all possible shifts to decrypt the message. The lack of complexity in the cipher makes it vulnerable to this type of straightforward attack.

MONO ALPHABETIC SUBSTITUTION CIPHER:

A Monoalphabetic Substitution Cipher is a type of substitution cipher where each letter of the plaintext is replaced by a corresponding letter in the ciphertext consistently throughout the entire message. In this cipher, a fixed substitution table is used, and each letter in the plaintext is replaced by the corresponding letter in the table.

For example, if we use a monoalphabetic substitution cipher with the following table:
Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext: XYZABCDEFGHIJKLMNOPQRSTUVW

Then the word "HELLO" would be encrypted as "EBIIL" using the substitution table above.
a monoalphabetic substitution cipher can be broken by a brute force attack. A brute force attack is an attempt to systematically try all possible keys until the correct one is found. In the case of a monoalphabetic substitution cipher, the key is the substitution table, which maps each letter of the alphabet to a corresponding letter in the ciphertext.

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

The reason why a brute force attack can be effective against a monoalphabetic substitution cipher is that there are only 26! (26 factorial) possible keys. Since each letter of the alphabet can be substituted with any other letter exactly once, the total number of possible keys is:
26! = 26 × 25 × 24 × ... × 3 × 2 × 1 ≈ 4.03 × 10^26
With sufficient computing power, a brute force attack can quickly test all possible keys and identify the correct one.

Frequency analysis can aid in breaking a monoalphabetic substitution cipher. Since the same plaintext letters are consistently replaced by the same ciphertext letters, patterns emerge in the frequency distribution of letters in the ciphertext. For example, the most frequent letter in the ciphertext is likely to represent the letter 'e' in the plaintext, which is the most common letter in the English language.

OUTPUT:

SHIFT CIPHER:

PART III

Plaintext:

my name is shreya

shift: 5

v Encrypt v | ^ Decrypt ^

Ciphertext

rd sfrj nx xmwjdf

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

PART III

Plaintext:

my name is shreya

shift: 10

v Encrypt v | ^ Decrypt ^

Ciphertext

wi xkwo sc crboik

MONOALPHABETIC SUBSTITUTION CIPHER:

Virtual Labs

Breaking the Mono-alphabetic Substitution Cipher

PART II

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

Scratchpad:

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

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 [ ] by plaintext character [ ] Modify

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

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

Your replacement history:

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

Virtual Labs

Breaking the Mono-alphabetic Substitution Cipher

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

Scratchpad:

```
YTxyvrQ 1 - qVHt vTr QxXXwv TVur: xuwYr wn XVQrq nwvvwtp Vt vTr
QwsrQXxto HwvT TrQ nwnvrQ, HTrt nTr tVvwYrn x vxuowtp, YuVvTrq HTwvr
QxXXwv HwvT x yVYorv HxvYT QEt yxnv. nTr WVuuVHn wv qVHt x QxXXwv TVur
HTrt nEqqrtuK nTr Wxuun x uVtp HxK vV x YEQwVEn Txuu HwvT RxtK uVYorq
qVVQn VW xuu nwmrn. nTr Wwtqn x nRxuu orK vV x qVVQ vVV nRxuu WVQ TrQ vV
Wwv, XEv vTQVEpT HTwYT nTr nrrn xt xvvQxYvwsr pxQqrt. nTr vTrt qwnYVsrQn
x XVvvur uxXruurq 'qQwto Rr', vTr YVtvrtvn VW HTwYT YxEnr TrQ vV nTQwto
vVV nRxuu vV QrxYT vTr orK. x Yxor HwvT 'rxv Rr' Vt wv YxEnrn TrQ vV
pQVH vV nEYT x vQrRrtqVEn nwmr TrQ Trxq Twvn vTr Yrwuwtp.
```

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 [1] by plaintext character [W] Modify

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

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

Your replacement history:

You replaced a by D You replaced a by D You replaced b by K You replaced c by X You replaced d by Y You replaced e by V You replaced f by R You replaced g by H You replaced h by Q You replaced i by E You replaced j by Q You replaced j by U You replaced k by T You replaced l by W

PART IV

Plaintext

```
welcome to the mystery text: when we speak plainly of
the mysteries which are ever around, both our spoken
and written words can lead us astray. but knowing a
mystery does not trouble the tongue as knowing a lie
does not trouble our mind. but these great many years
we have been told of this lie by those whom we trusted
most. but surely we must trust 'the one who', the
creator of all things. the next chapter in history is
```

key = wjexcvtqkurmnopqglvwfxdyzs Generate Random Key

v Encrypt v ^ Decrypt ^

☐ Remove Punctuation

Ciphertext

```
dcmepnc wp wqc nzvwclz wcyw: dqco dc vqcwr qmwkomz
pv wqc nzvwclkcv dqkeq wlc cxcl wlpfox, jpwq pfl
vqprco wox dlkwwco dplxv ewo mcwx fv wvwlwz. jfw
ropdkot w nzvwclz xpcv opw wlpfjmc wqc wpotfc wv
ropdkot w mkc xpcv opw wlpfjmc pfl nkox. jfw wqcvc
tlcww nwoz zcwlv dc qwxc jcco wpmx pv wqkv mkc jz
```

CONCLUSION:

Hence, we have illustrated symmetric cryptography by implementing classical ciphers like the shift cipher and mono-alphabetic substitution cipher.

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

LAB ASSIGNMENT NO. 2

AIM: Cryptanalysis or decoding of polyalphabetic ciphers: Playfair, Vigenere cipher.

LAB OUTCOME ATTAINED:

LO1: Illustrate symmetric cryptography by implementing classical ciphers.

THEORY:

VIGENERE CIPHER

The Vigenère cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. It uses a keyword (or keyphrase) to determine the shift value for each letter in the plaintext. The keyword is repeated as necessary to match the length of the plaintext.
In the Vigenère cipher, you use a keyword (or keyphrase) to determine the shift value for each letter in your plaintext. First, choose a keyword that you'll repeat to match the length of your plaintext.

For example, let's say your keyword is "KEY" and your plaintext is "HELLO" (all in uppercase).

Keyword: "KEYKEY" (repeating the keyword to match the length of the plaintext).
Plaintext: "HELLO."
Next, refer to the Vigenère Table (or Vigenère Square).

Encrypt your plaintext:

- Match each letter of your plaintext with the corresponding letter of your keyword (H -> K, E -> E, L -> Y, L -> K, O -> E).
- Find the corresponding letter in the Vigenère table at the intersection of the row and column of the matching letters.
- Your encrypted ciphertext is "KYKYE."

USE OF KASISKI TEST

The Kasiski test is a clever method to break the Vigenère cipher by exploiting repeated patterns in the ciphertext. By detecting these repeating substrings and measuring the distances between them, the test can infer the probable length of the keyword used in the encryption. Finding common factors among these distances helps to narrow down the potential keyword lengths.

Once the likely keyword length is determined, the ciphertext is divided into groups based on this length. Each group is then analysed separately using frequency analysis, as if it were encrypted with a simple substitution cipher. Since each group corresponds to a different shift value in the Vigenère cipher, frequency analysis becomes more effective.

By calculating the shift values for each group, the test reconstructs parts of the keyword. By piecing together these parts, the complete keyword is obtained, allowing for the decryption of the entire ciphertext.

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

Although the Kasiski test can be quite effective, its success depends on the presence of repeating patterns in the ciphertext and the length of the keyword. For shorter ciphertext or longer keywords, the test's accuracy may decrease, and additional techniques might be required for decryption.

PLAYFAIR CIPHER

The Playfair cipher is a digraphic substitution cipher used to encrypt plaintext. It operates on pairs of letters (digraphs) instead of individual letters, making it more secure than simple substitution ciphers. The cipher uses a 5x5 matrix (Playfair square) of letters, typically excluding "J," to create the encryption key.

In the Playfair cipher example, the keyword "KEYWORD" is used to generate the Playfair square. The plaintext "HELLO WORLD" is preprocessed into digraphs ("HE LX LO WO RL DX"). Applying encryption rules, the digraphs are encrypted: "HE" becomes "EK," "LX" becomes "RC," "LO" becomes "OD," and "WO" becomes "BM." The final ciphertext is "EKRCOMEDY." To decrypt, both sender and receiver must use the same Playfair square with the shared keyword to reverse the process and retrieve the original plaintext "HELLOWORLD." The cipher's digraphic approach enhances security compared to simple substitution ciphers, making it a valuable historical encryption technique.

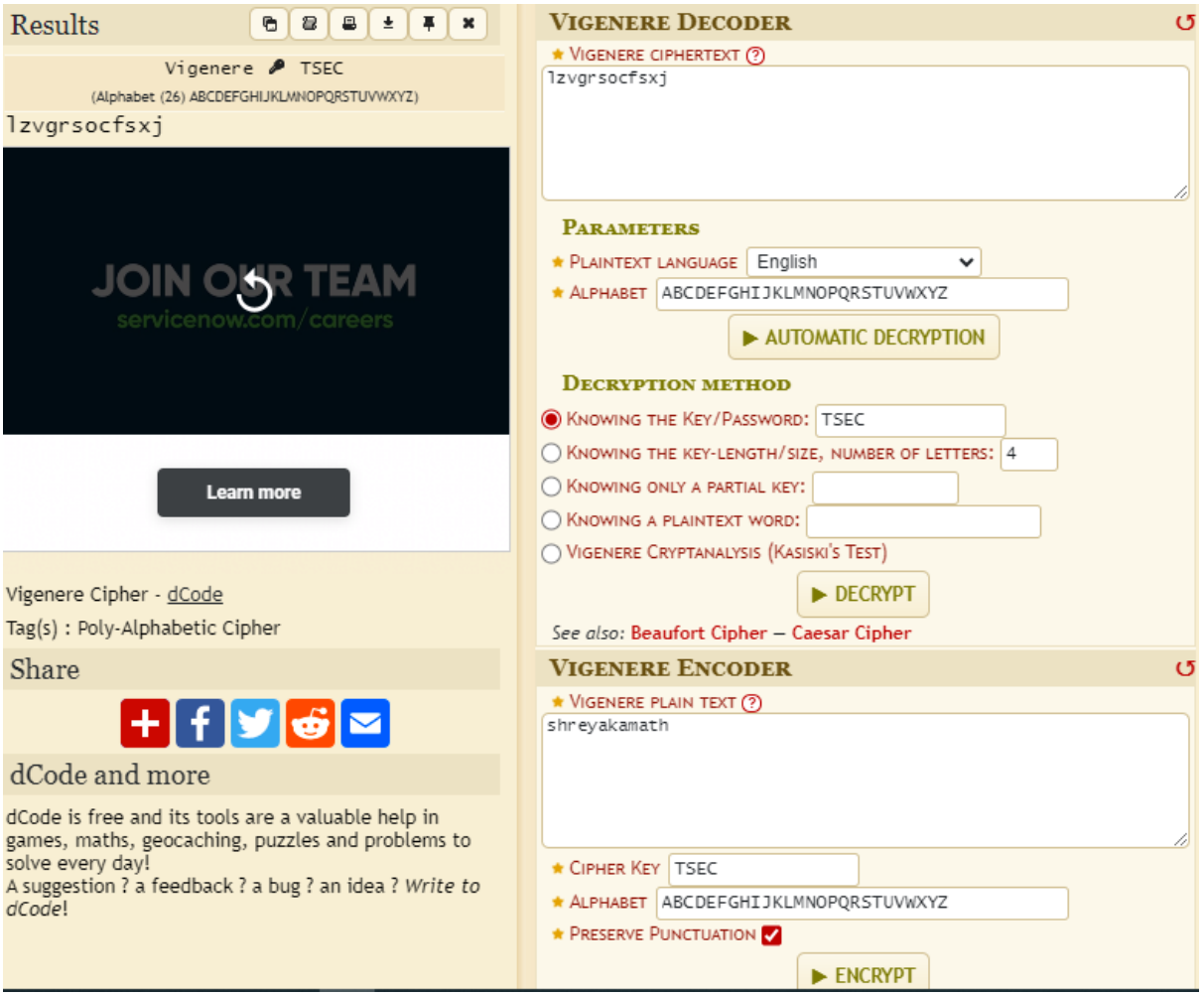
CRYPTANALYSIS OF PLAYFAIR CIPHER

Cryptanalysis of the Playfair cipher aims to decrypt the ciphertext without knowing the keyword or Playfair square. Techniques like frequency analysis, pattern recognition, and the Kasiski examination are employed to identify repeating patterns and deduce likely digraphs. Brute force attacks involve trying all possible keyword and square combinations. Known and chosen plaintext attacks use partial knowledge of the plaintext-ciphertext pairs to infer the key. The Playfair cipher's security depends on the keyword's complexity and the ciphertext length. While it offers better security than simple substitution ciphers, cryptanalysis methods can still be effective with sufficient ciphertext data and clever analysis.

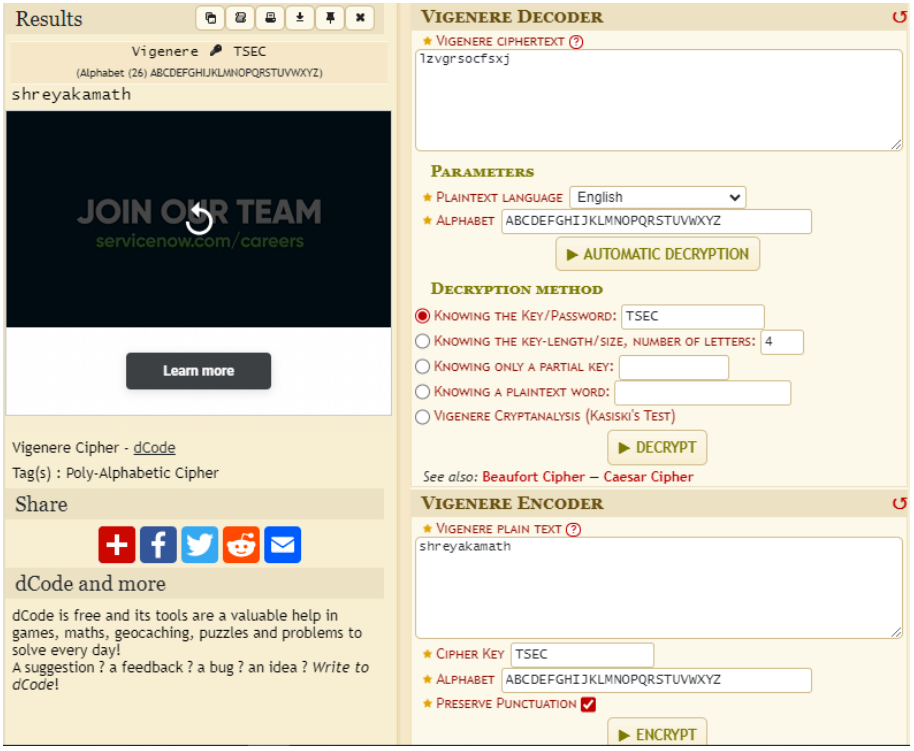
Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

OUTPUT:

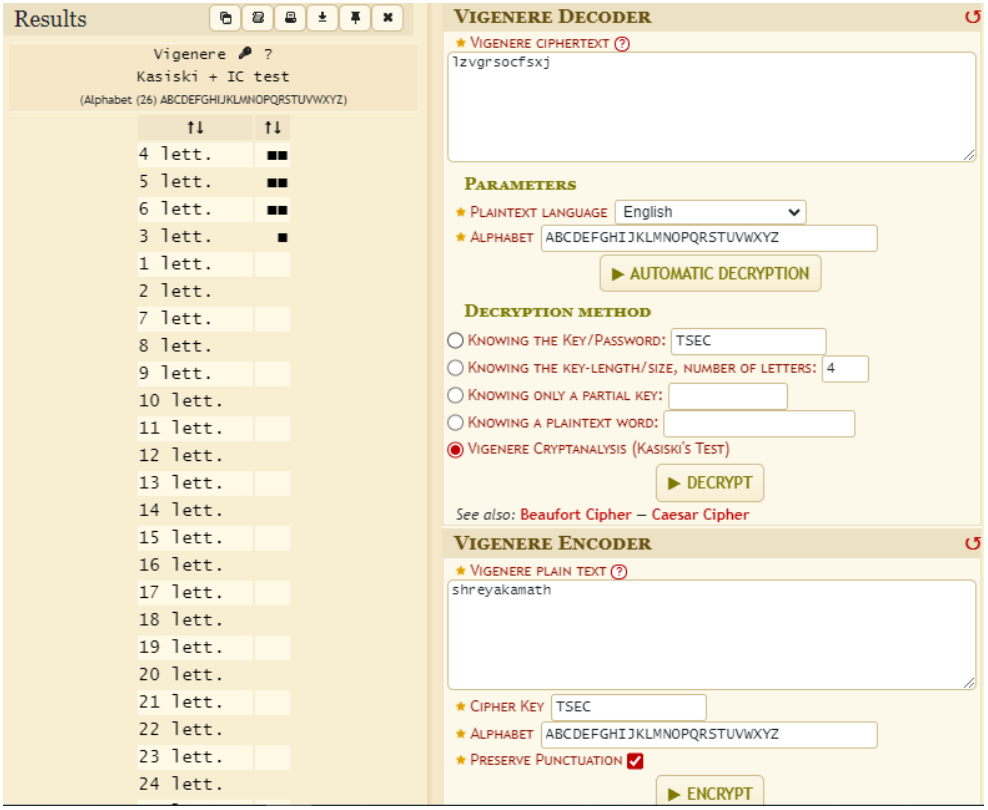
VIGENERE CIPHER

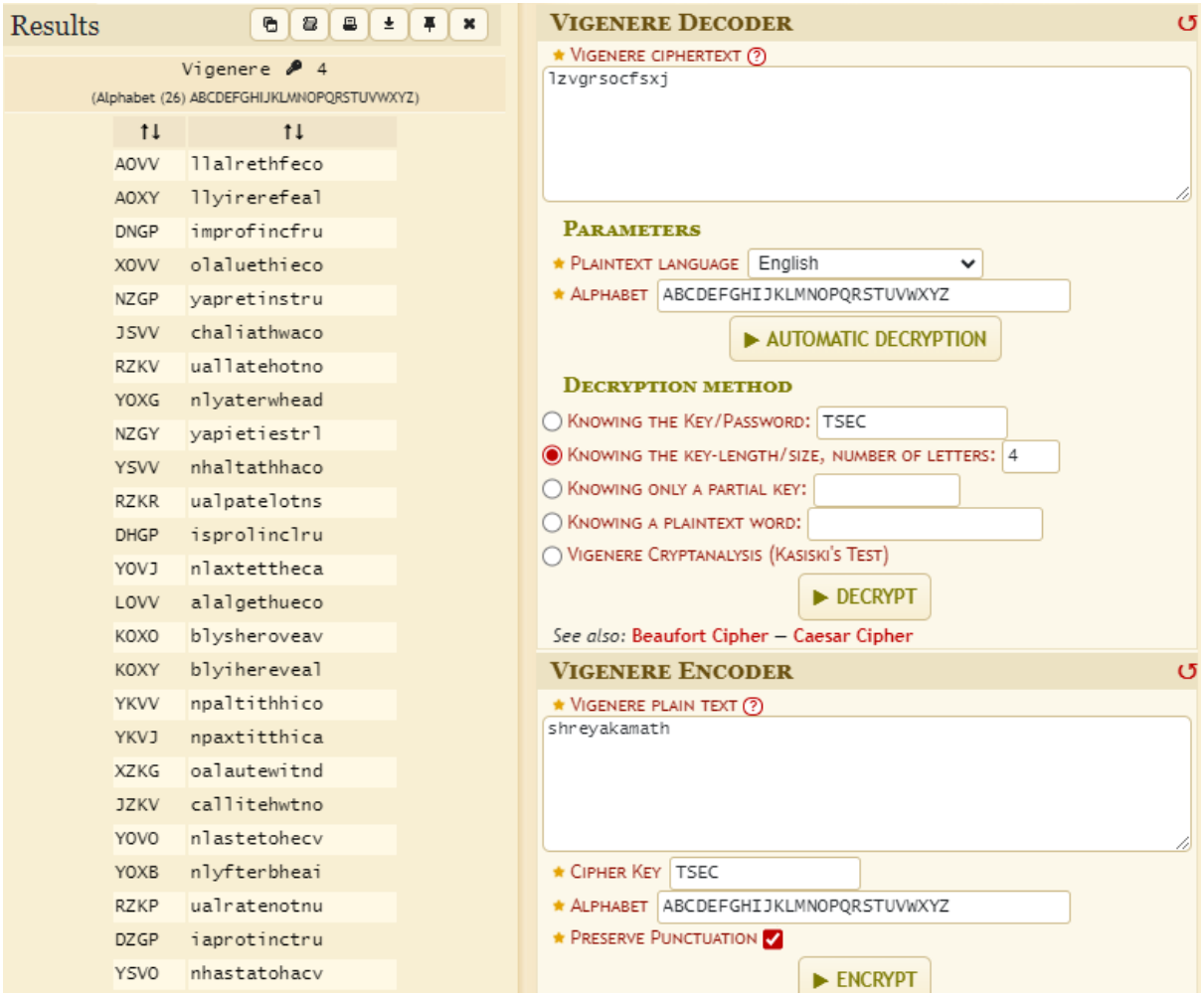


Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.





Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

Results

Vigenere 🔑 TS

(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

TS shcoyavkmaer

Vigenere Cipher - dCode

Tag(s) : Poly-Alphabetic Cipher

VIGENERE DECODER

★ VIGENERE CIPHERTEXT

lzvgrsocfsxj

PARAMETERS

★ PLAINTEXT LANGUAGE English

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

▶ AUTOMATIC DECRYPTION

DECRYPTION METHOD

- ○ KNOWING THE KEY/PASSWORD: TSEC
- ○ KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS: 4
- ◉ KNOWING ONLY A PARTIAL KEY: TS
- ○ KNOWING A PLAINTEXT WORD:
- ○ VIGENERE CRYPTANALYSIS (KASISKI'S TEST)

▶ DECRYPT

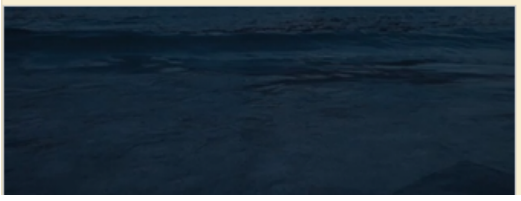
See also: Beaufort Cipher – Caesar Cipher

Results

Vigenere 🔑 ?

(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

↑↓	↑↓
TSEC	shreyakamath
HLBUXA	eoumushreyaj
WVOOZJ	pehssjshreya
XYHSZL	obooshreyayy
KEFOKB	bvqshreyaeni
OHOPNU	xshreyavrdkp
QCDZAO	vxshreyactxv
TSECTS	shreyavkbqer
#8	



VIGENERE DECODER

★ VIGENERE CIPHERTEXT

lzvgrsocfsxj

PARAMETERS

★ PLAINTEXT LANGUAGE English

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

▶ AUTOMATIC DECRYPTION

DECRYPTION METHOD

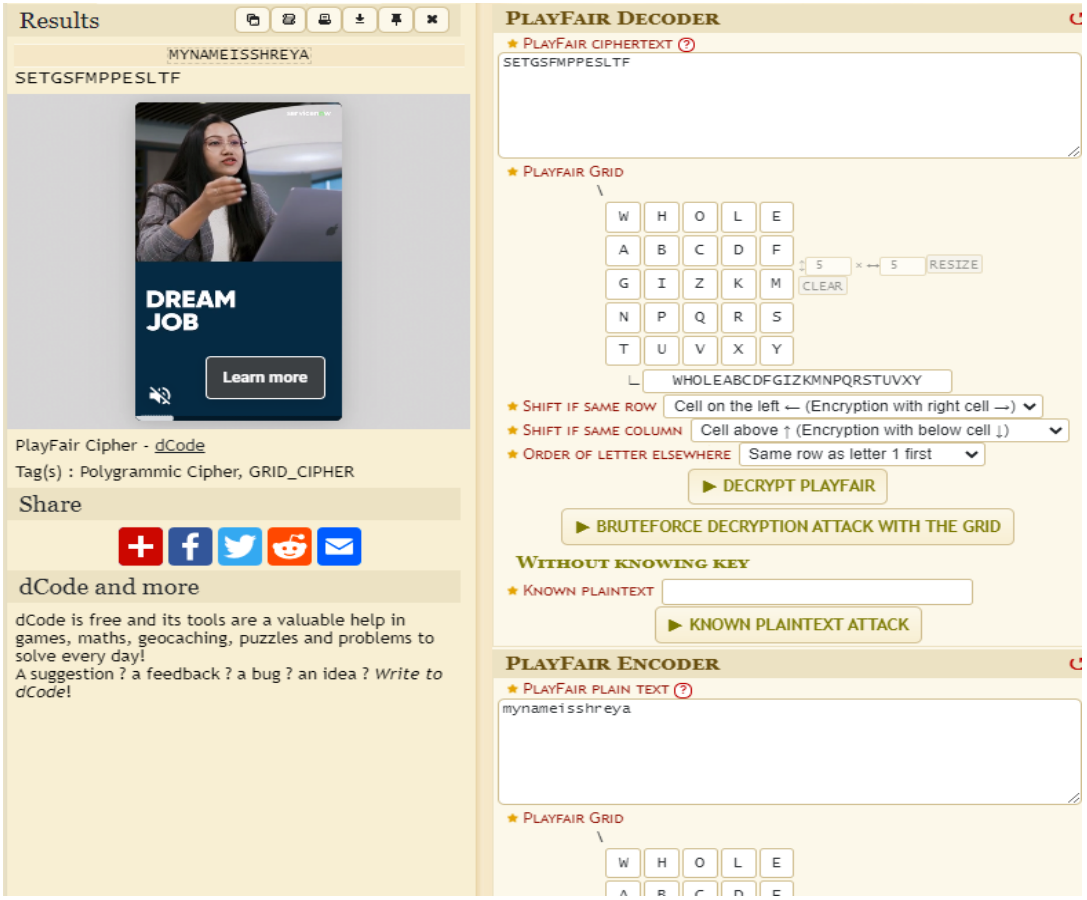
- ○ KNOWING THE KEY/PASSWORD: TSEC
- ○ KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS: 4
- ○ KNOWING ONLY A PARTIAL KEY:
- ◉ KNOWING A PLAINTEXT WORD: SHREYA
- ○ VIGENERE CRYPTANALYSIS (KASISKI'S TEST)

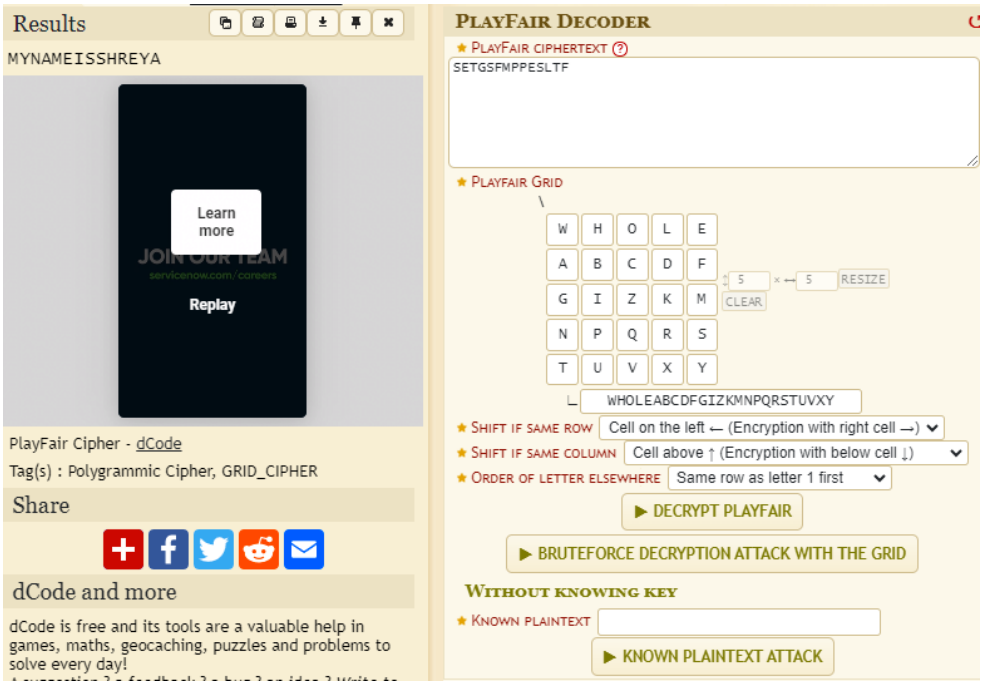
▶ DECRYPT

See also: Beaufort Cipher – Caesar Cipher

Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

PLAYFAIR CIPHER





Roll no. : 53
Name: Shreya Kamath
Date: 27th July, 2023.

CONCLUSION:

Hence, we have conducted Cryptanalysis or decoding of polyalphabetic ciphers such as the Playfair and Vigenere cipher. We also understood how the Kasiski Test is used to break the Vigenere cipher.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

LAB ASSIGNMENT NO. 3

AIM: To study Block cipher modes of operation using Advanced Encryption Standard (AES).

LAB OUTCOME ATTAINED:

LO 2: Demonstrate Key management, distribution and user authentication.

THEORY:

AES (Advanced Encryption Standard) is a symmetric-key encryption algorithm, meaning the same key is used for both encryption and decryption. It's a widely adopted encryption standard for securing sensitive data and is known for its efficiency and security.

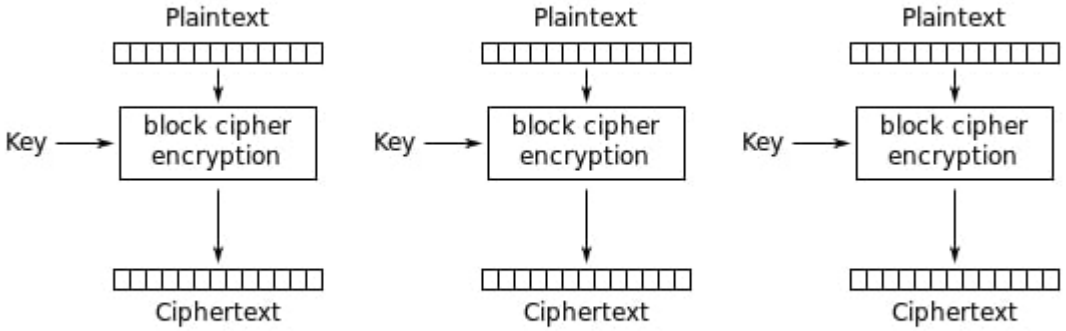
1. **Type of Cipher**: AES is a block cipher, which means it encrypts data in fixed-size blocks (128 bits or 16 bytes) rather than one bit at a time.
2. **Number of Rounds**: The number of rounds in AES depends on the key size. For AES-128, there are 10 rounds; for AES-192, there are 12 rounds; and for AES-256, there are 14 rounds. Each round involves a series of operations.
3. **Key Size**: AES supports key sizes of 128, 192, or 256 bits. The key size determines the security level, with longer keys providing stronger encryption.
4. **Block Size:** AES operates on data blocks of 128 bits (16 bytes). This block size remains fixed regardless of the key size.
5. **Operations in Each Round**: In each round of AES, several operations are performed on the data, including:
    - SubBytes: Non-linear substitution where each byte in the block is replaced with a corresponding byte from a fixed table (called the S-box).
    - ShiftRows: Bytes in each row of the block are shifted left by different offsets.
    - MixColumns: A mathematical mixing operation is performed on the columns of the block.
    - AddRoundKey: The block is XORed with a portion of the expanded encryption key derived from the original encryption key.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

MODES OF OPERATION:

ECB Mode (Electronic Codebook Mode):

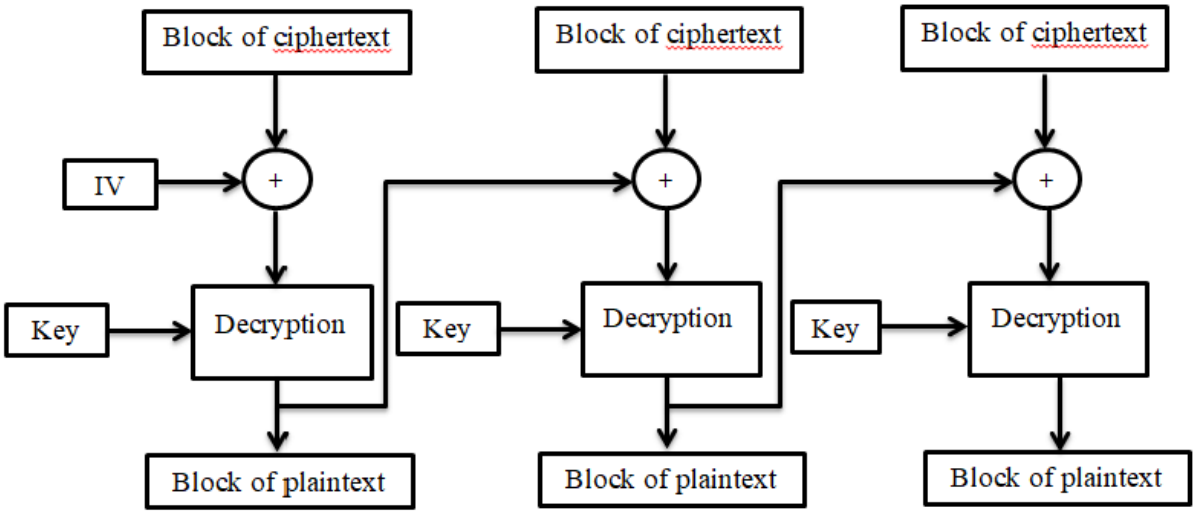
In ECB mode, each block of plaintext is encrypted independently with the same encryption key. This means that identical blocks of plaintext will result in identical blocks of ciphertext, which can be a security vulnerability.



Electronic Codebook (ECB) mode encryption

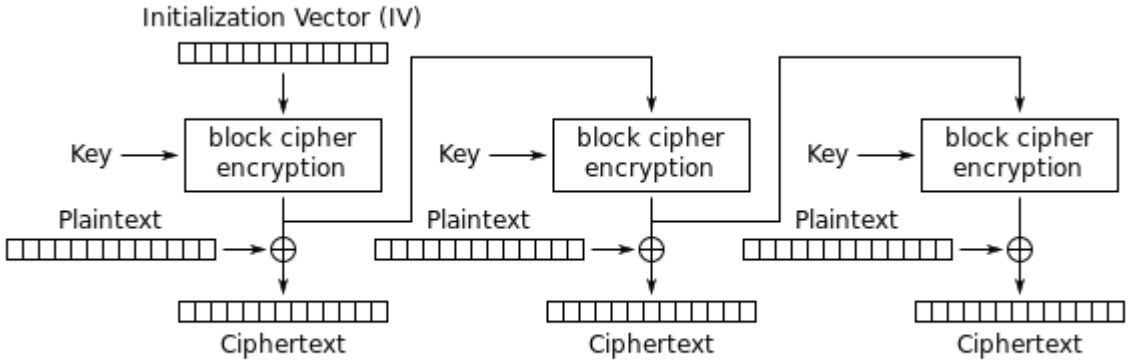
CBC Mode (Cipher Block Chaining Mode):

In CBC mode, each block of plaintext is XORed with the previous ciphertext block before encryption. This "chaining" of blocks adds complexity and ensures that identical plaintext blocks do not produce identical ciphertext blocks.



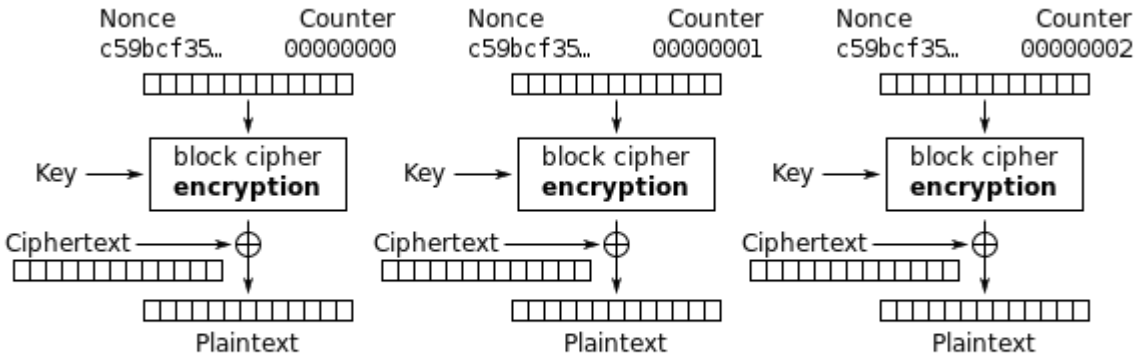
Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

OFB Mode (Output Feedback Mode):
OFB mode converts the block cipher into a stream cipher. It generates a stream of pseudo-random bits using an initialization vector (IV) and the encryption key. This stream is then XORed with the plaintext to produce ciphertext.



Output Feedback (OFB) mode encryption

Counter Mode (CTR Mode):
Counter mode also converts the block cipher into a stream cipher. It uses a counter value as an IV, which is incremented for each block. The counter value is encrypted to produce a keystream, which is then XORed with the plaintext.



Counter (CTR) mode decryption

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

OUTPUT:

PART I

Choose your mode of operation: Electronic Code Book (ECB)

PART II

Key size in bits: 128

Plaintext:
```
38446fd9 643b8207 90f83cd8 927f25ef
1e01bc42 356c2ffa d87caf05 5ebebdf9
2bd7d18e 54e31db7 847c41c3 a78c27db
eb48922a cb1bef7d 261c959e ae006c05
a7280a41 9a6405ee b439bf3d 64a038aa
```
Next Plaintext Key:

```
1a5951c8 8cbecfff 6ba85166 51b914eb
```
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

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

AES and Modes of Operation

Key size in bits: 128

Plaintext: 38446fd9 643b8207 90f83cd8 927f25ef
1e01bc42 356c2ffa d87caf05 5ebebdf9
2bd7d18e 54e31db7 847c41c3 a78c27db
eb48922a cb1bef7d 261c959e ae006c05
a7280a41 9a6405ee b439bf3d 64a038aa
Next Plaintext Key: 1a5951c8 8cbecfff 6ba85166 51b914eb Next Keytext

IV: Next IV

CTR: Next CTR

PART III

Calculate XOR:

Calculate XOR

XOR:

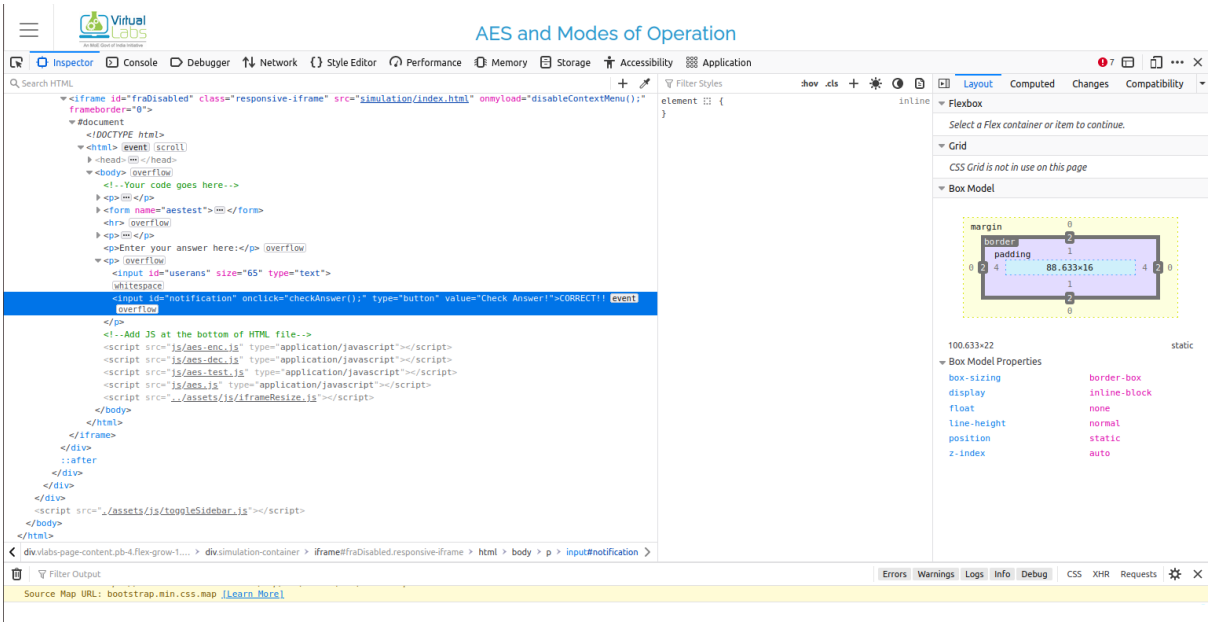
PART IV

Key in hex: 1a5951c8 8cbecfff 6ba85166 51b914eb
Plaintext in hex: 38446fd9 643b8207 90f83cd8 927f25ef
Ciphertext in hex: f7a052ca 81711404 1e47f52a 7761f7ea
Encrypt Decrypt Clear

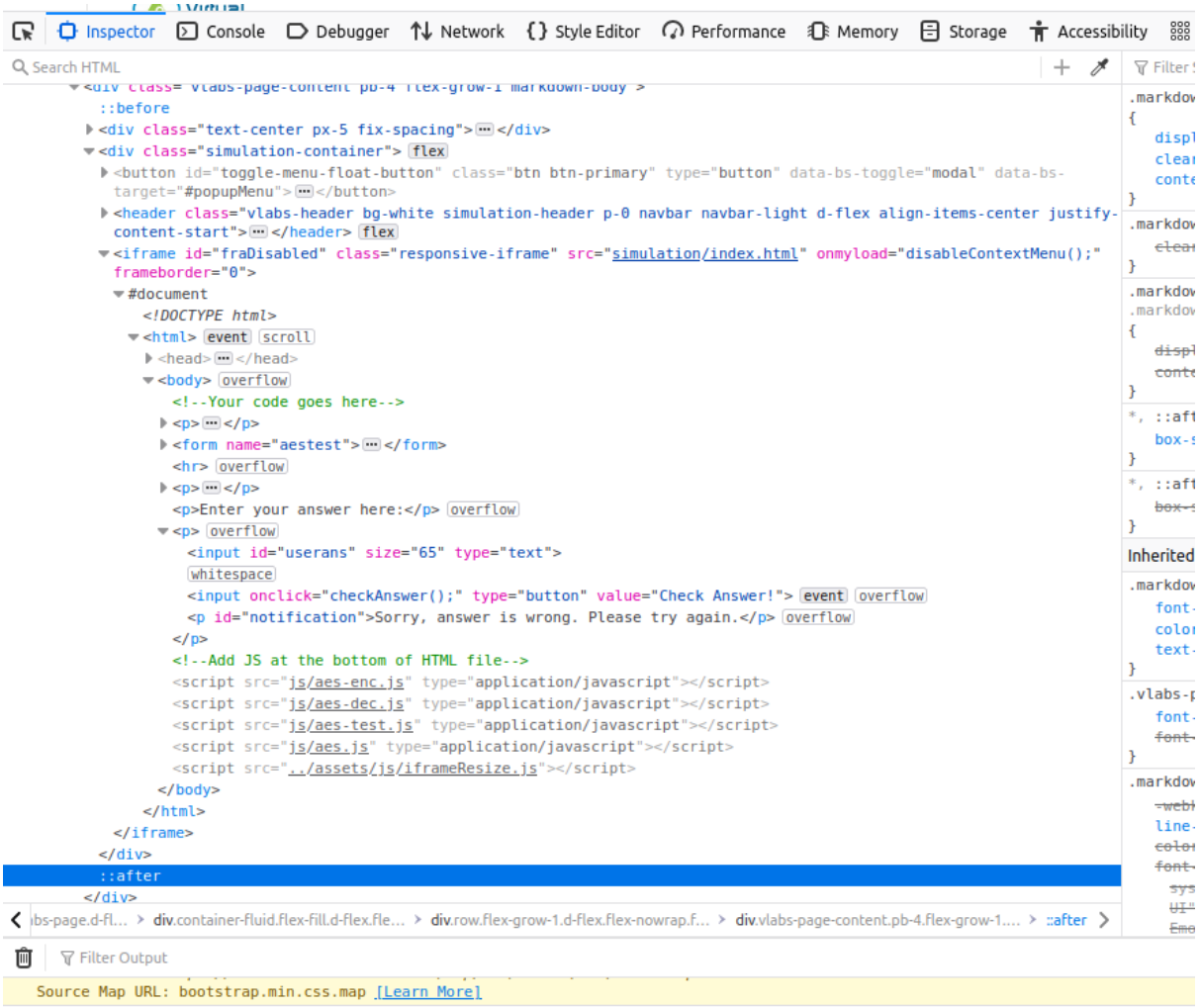
PART V

Enter your answer here:

f7a052ca 81711404 1e47f52a 7761f7ea Check Answer!



Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

AES and Modes of Operation

PART I

Choose your mode of operation: Cipher Block Chaining

PART II

Key size in bits: 128

Plaintext:
```
9335300b e2435f9b 7286bd7c 7701d9be
15c7c6c4 c3c32ab2 98d17dfb a1d2b395
584a0e79 083b535a 5b37585a 80576f4e
7863efc4 ed353c4b 2de3520b eb7f4ec1
1414bbfe 19c8c67f 573a51c7 5e9257bb
```
Next Plaintext Key: a90a0373 4ecfdcc5 37f24b3f b6abe197 Next Keytext

IV: 963c73b8 ded48600 49cb817d a3939614 Next IV

PART III

Calculate XOR:

1414bbfe 19c8c67f 573a51c7 5e9257bb

963c73b8 ded48600 49cb817d a3939614 Calculate XOR

XOR: 8228c846 c71c407f 1ef1d0ba fd01c1af

PART IV

Key in hex: a90a0373 4ecfdcc5 37f24b3f b6abe197

Plaintext in hex: 8228c846 c71c407f 1ef1d0ba fd01c1af

Ciphertext in hex: c1b2a50f cba4a942 2d1c8dd5 0a19d0cf

Encrypt Decrypt Clear

AES and Modes of Operation

Key size in bits: 128

Plaintext:
```
9335300b e2435f9b 7286bd7c 7701d9be
15c7c6c4 c3c32ab2 98d17dfb a1d2b395
584a0e79 083b535a 5b37585a 80576f4e
7863efc4 ed353c4b 2de3520b eb7f4ec1
1414bbfe 19c8c67f 573a51c7 5e9257bb
```
Next Plaintext Key: a90a0373 4ecfdcc5 37f24b3f b6abe197 Next Keytext

IV: 963c73b8 ded48600 49cb817d a3939614 Next IV

PART III

Calculate XOR:

1414bbfe 19c8c67f 573a51c7 5e9257bb

963c73b8 ded48600 49cb817d a3939614 Calculate XOR

XOR: 8228c846 c71c407f 1ef1d0ba fd01c1af

PART IV

Key in hex: a90a0373 4ecfdcc5 37f24b3f b6abe197

Plaintext in hex: 8228c846 c71c407f 1ef1d0ba fd01c1af

Ciphertext in hex: c1b2a50f cba4a942 2d1c8dd5 0a19d0cf

Encrypt Decrypt Clear

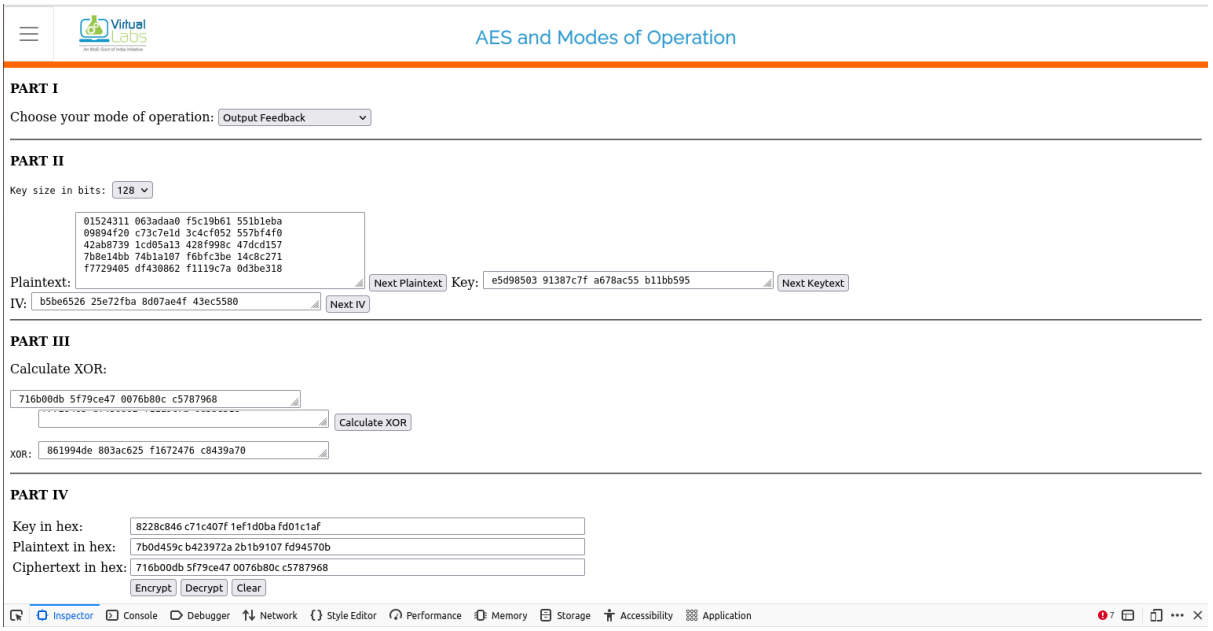
PART V

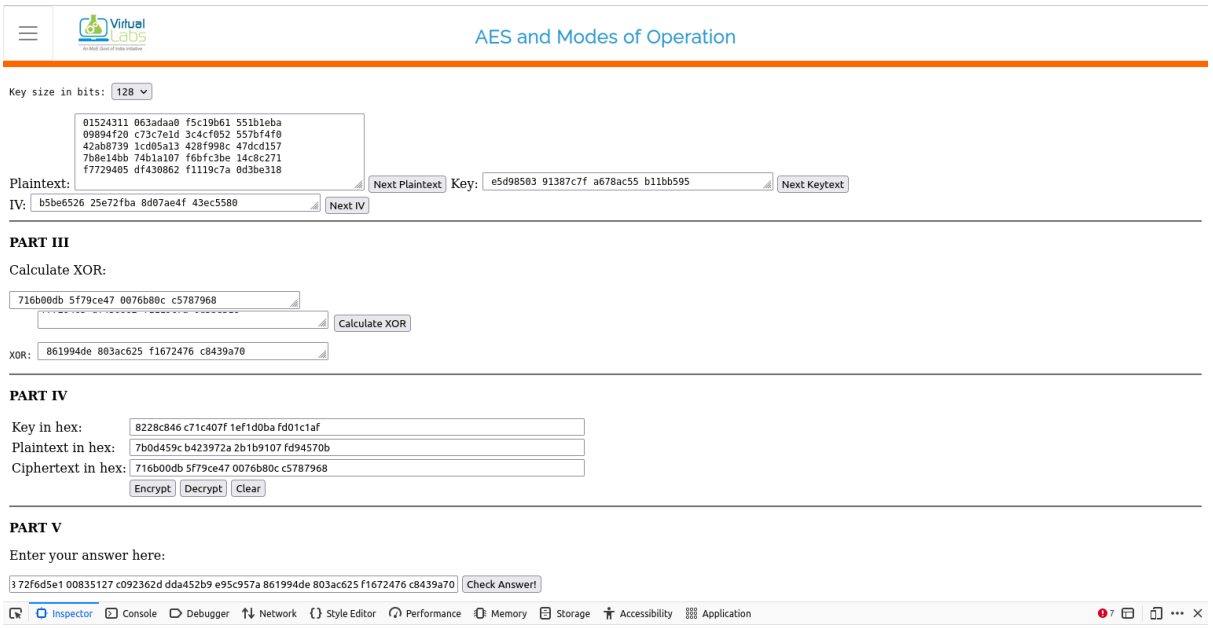
Enter your answer here:

i31 d9be843f a404bf30 def90712 715fd1de d10ae8e5 8228c846 c71c407f 1ef1d0ba fd01c1af Check Answer!

Sorry, answer is wrong. Please try again.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.





CONCLUSION:

Hence, I have understood the concept of AES encryption standard algorithm and its various modes and performed encryption and decryption using various modes on a virtual simulator.

Roll no. : 53
Name: Shreya Kamath
Date: 31st July, 2023.

LAB ASSIGNMENT NO. 4

AIM: Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA.

LAB OUTCOME ATTAINED:

LO 2: Demonstrate Key management, distribution and user authentication.

THEORY:

RSA

RSA is a widely used public-key encryption algorithm. It involves key generation with two large prime numbers, calculating the modulus and totient, and choosing public and private exponents. The security relies on the difficulty of factoring the large modulus, ensuring secure communication and data encryption. The public key is used for encryption, while the private key is used for decryption.

Steps for Key Generation in RSA

Step 1: Choose Two Large Prime Numbers
Select two distinct prime numbers, typically denoted as "p" and "q." These prime numbers should be large to enhance the security of the RSA key. The product of "p" and "q" is used to calculate the modulus "n" ($n = p * q$).

Step 2: Calculate the Modulus "n"
Compute the modulus "n" by multiplying the two selected prime numbers: $n = p * q$.

Step 3: Calculate the Totient of "n" ($\varphi(n)$)
The totient of "n," denoted as $\varphi(n)$, is calculated as $\varphi(n) = (p-1) * (q-1)$. The totient function counts the number of positive integers that are coprime (relatively prime) to "n."

Step 4: Choose the Public Exponent (e)
Select a small public exponent "e" (usually a prime number), where $1 < e < \varphi(n)$, and "e" is coprime with $\varphi(n)$ (i.e., $\gcd(e, \varphi(n)) = 1$). The public key is represented by (e, n).

Step 5: Calculate the Private Exponent (d)
Compute the private exponent "d" such that $(d * e) \% \varphi(n) = 1$. In other words, "d" is the modular multiplicative inverse of "e" modulo $\varphi(n)$. The private key is represented by (d, n).

Step 6: Public and Private Key Generation
The generated public key is (e, n), and the corresponding private key is (d, n).

The security of RSA relies on the difficulty of factoring the large modulus "n" into its prime factors "p" and "q." The larger the prime numbers used, the more secure the RSA key. The public key (e, n) is used for encryption, while the private key (d, n) is kept secret and used for decryption.

Roll no. : 53
Name: Shreya Kamath
Date: 31st July, 2023.

DIGITAL SIGNATURE

A digital signature is a cryptographic technique used to verify the authenticity and integrity of digital messages or documents. It involves the use of a private key to encrypt a unique hash of the message, creating a digital signature. The recipient can then use the corresponding public key to decrypt the signature and verify the message's origin and content.

Digital Signature Generation Process:

1. Hashing: The signer creates a hash (fixed-size digital fingerprint) of the message using a cryptographic hash function (e.g., SHA-256). This produces a unique representation of the message.

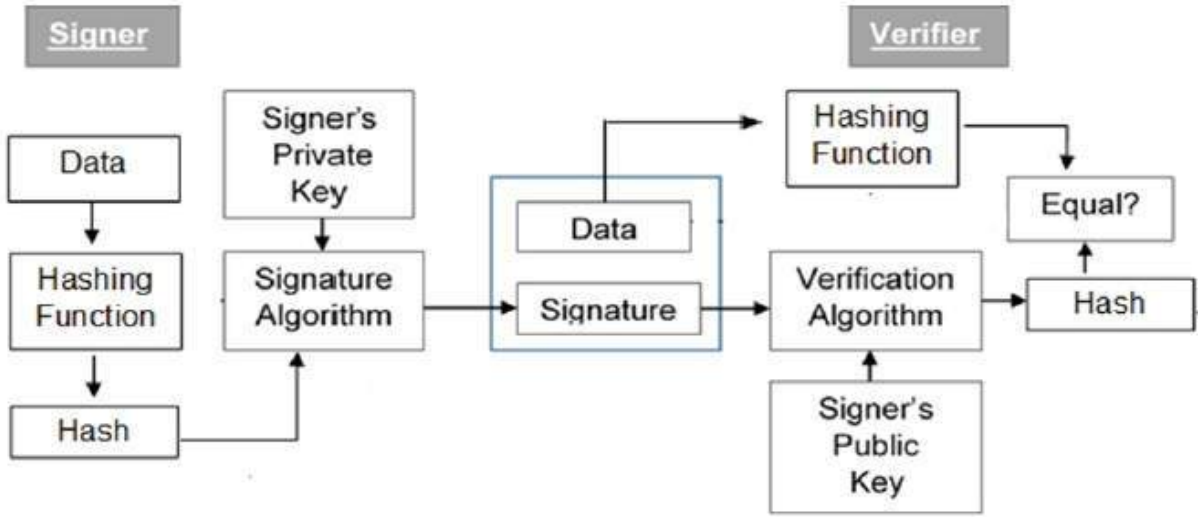
2. Private Key Encryption: The signer encrypts the hash using their private key (from their public-private key pair) to create the digital signature. This ensures that only the signer's private key can produce the signature for that specific message.

Digital Signature Verification Process:

1. Hashing: The recipient of the message computes the hash of the received message using the same cryptographic hash function used by the signer.

2. Public Key Decryption: The recipient decrypts the received digital signature using the signer's public key (obtained from a trusted source like a digital certificate).

3. Comparison: The recipient compares the computed hash with the decrypted signature. If they match, it confirms the integrity and authenticity of the message, as only the signer's private key could have produced the matching signature for that specific message.



By using digital signatures, the recipient can verify the origin and integrity of the message, ensuring that it has not been altered in transit and came from the claimed sender.

Roll no. : 53
Name: Shreya Kamath
Date: 31st July, 2023.

OUTPUT:
RSA

Plaintext (string):

shreya kamath
encrypt

Ciphertext (hex):

```
38c8cd63b594936af48200940978d7dd719edcbf42a91f7775da45f33b26d6b5
4d7cc0ef5b060bd188b79eec06022ee0aab1e2e6e5baf155497b7cae007d64d0
```

decrypt

Decrypted Plaintext (string):
shreya kamath

Status:
Decryption Time: 1ms

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 512

Modulus (hex):

```
BC86E3DC782C446EE756B874ACECF2A115E613021EAF1ED5EF295BEC2BED899D
26FE2EC896BF9DE84FE381AF67A7B7CBB48D85235E72AB595ABF8FE840D5F8DB
```

Public exponent (hex, F4=0x10001):
3

Private exponent (hex):

```
7daf4292fac82d9f44e47af87348a1c0b9440cac1474bf394a1b929d729e5bbc
f402f29a9300e11b478c091f7e5dacd3f8edae2effe3164d7e0eeada87ee817b
```

P (hex):

```
ef3fc61e21867a900e01ee4b1ba69f5403274ed27656da03ed88d7902cce693f
```

Roll no. : 53
Name: Shreya Kamath
Date: 31st July, 2023.

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 512

Modulus (hex):

```
BC86E3DC782C446EE756B874ACECF2A115E613021EAF1ED5EF295BEC2BED899D
26FE2EC896BF9DE84FE381AF67A7B7CBB48D85235E72AB595ABF8FE840D5F8DB
```

Public exponent (hex, F4=0x10001):

```
3
```

Private exponent (hex):

```
7daf4292fac82d9f44e47af87348a1c0b9440cac1474bf394a1b929d729e5bbc
f402f29a9300e11b478c091f7e5dacd3f8edae2effe3164d7e0eeada87ee817b
```

P (hex):

```
ef3fc61e21867a900e01ee4b1ba69f5403274ed27656da03ed88d7902cce693f
```

Q (hex):

```
c9b9fcc298b7d1af568f85b50e749539bc01b10a68472fe1302058104821cd65
```

D mod (P-1) (hex):

```
9f7fd9696baefc6009569edcbd19bf8d576f89e1a439e6ad4905e50ac8899b7f
```

D mod (Q-1) (hex):

```
867bfdd7107a8bca39b503ce09a30e267d567606f02f7540cac03ab5856bde43
```

1/Q mod P (hex):

```
412d6b551d93ee1bd7dccafc63d7a6d031fc66035ecc630ddf75f949a378cd9d
```

DIGITAL SIGNATURE

Roll no. : 53
Name: Shreya Kamath
Date: 31st July, 2023.

Plaintext (string):

shreyakamath SHA-1

Hash output(hex):

7b1da690f567252df0b030a76b9f8f0e3b8b7771

Input to RSA(hex):

7b1da690f567252df0b030a76b9f8f0e3b8b7771 Apply RSA

Digital Signature(hex):

```
09e02be0837d0407bd718f959df06efaf3e882910f20a99016349a47d2e78b85
969b45e00cecc8f0902900f552f518f22857ce411e301a08810d6588e1ba1ef1
```

Digital Signature(base64):

```
CeAr4IN9BAe9cY+VnfBu+vPogpEPIKmQFjSaR9Lni4WWm0XgDOzI8JApAPVS9Rjy
KFfOQR4wGgiBDWWI4boe8Q==
```

Status:

Time: 2ms

RSA public key

Public exponent (hex, F4=0x10001):

3

Modulus (hex):

```
BC86E3DC782C446EE756B874ACECF2A115E613021EAF1ED5EF295BEC2BED899D
26FE2EC896BF9DE84FE381AF67A7B7CBB48D85235E72AB595ABF8FE840D5F8DB
```

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3)

CONCLUSION:

Hence, we successfully implemented the RSA cryptosystem and the digital signature scheme. Through the secure key management and authentication processes, we established a robust encryption system and a reliable method to verify message authenticity, ensuring secure communication and data integrity.

Roll no. : 53
Name: Shreya Kamath
Date: 17th August, 2023.

LAB ASSIGNMENT NO. 5

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

LAB OUTCOME ATTAINED:

LO 2: 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`
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.

Roll no. : 53
Name: Shreya Kamath
Date: 17th August, 2023.

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`

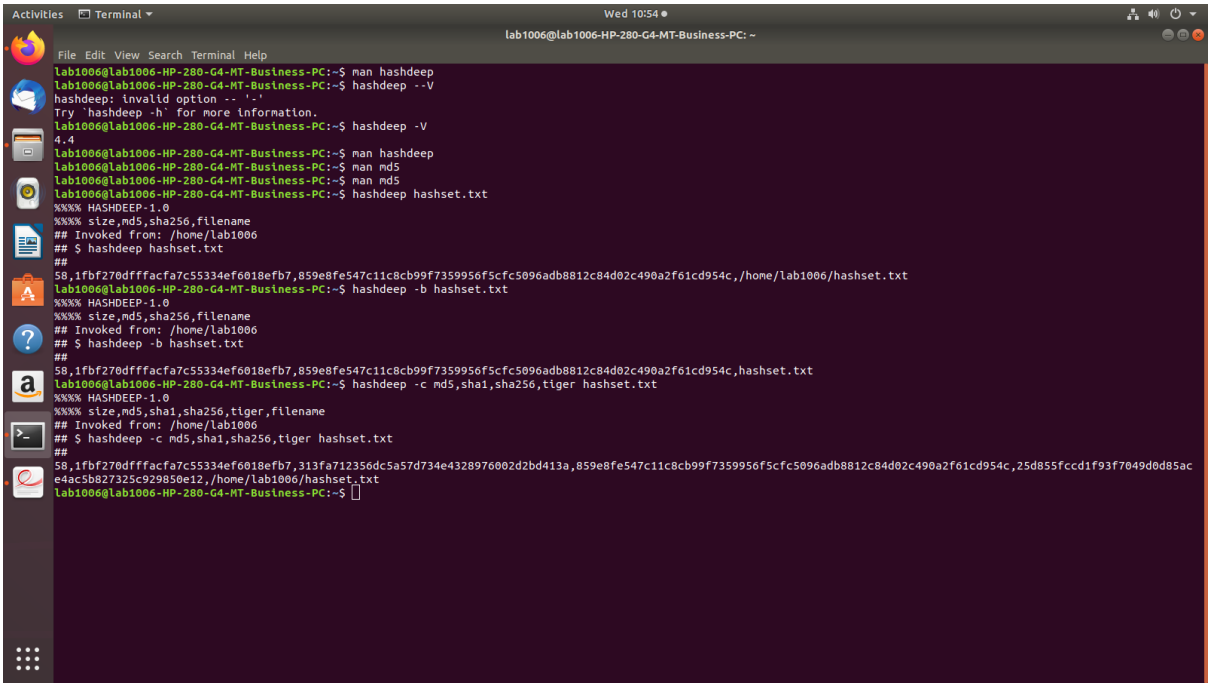
21. Audit After Renaming File:
  - Rename File: `mv /home/user/myfiles/shreya.txt /home/user/myfiles/backup.txt`
  - Audit Again: `hashdeep -v -a -r -k hashset1.txt /home/user/myfiles`

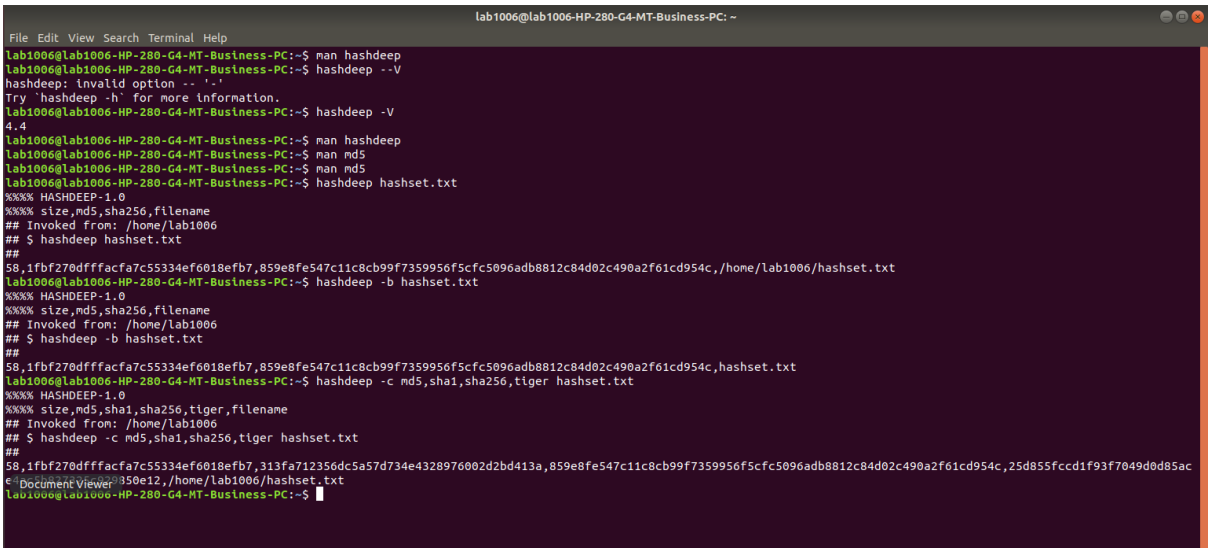
22. Verbose Audit Output:
  - More Verbose: `hashdeep -vv -a -r -k hashset1.txt /home/user/myfiles`
  - Very Verbose: `hashdeep -vvv -a -r -k hashset1.txt /home/user/myfiles`

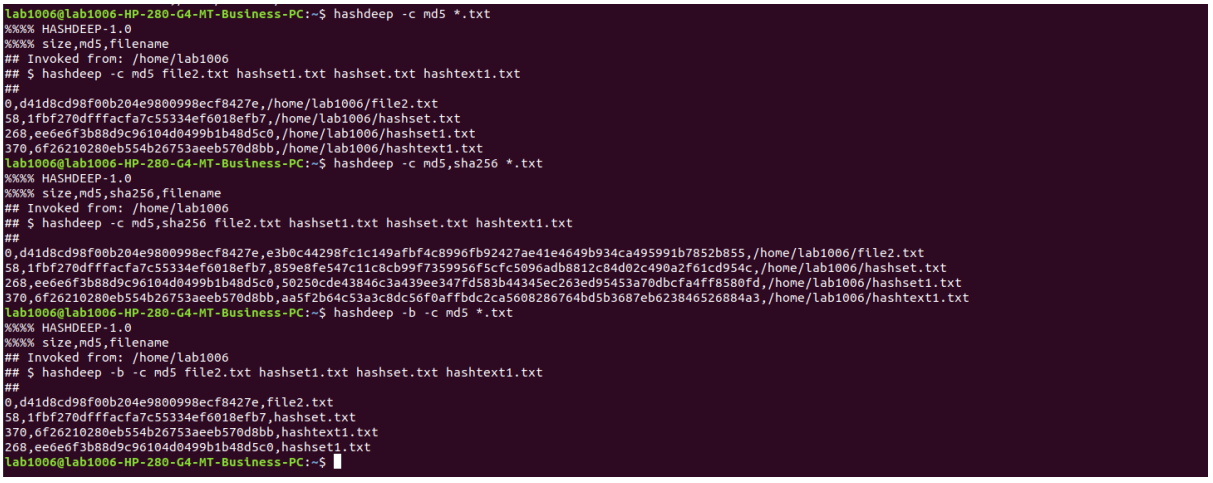
Note: Replace the paths and filenames with actual directory and file names as needed.

OUTPUT:

Roll no. : 53
Name: Shreya Kamath
Date: 17th August, 2023.







Roll no. : 53
Name: Shreya Kamath
Date: 17th August, 2023.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 hashset1.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 -p 100 hashset1.txt
##
100,52c24a1f83621f9e5be6114f45b1581f,/home/lab1006/hashset1.txt offset 0-99
100,27ea0f4cfb65783f4a79db493f28914b,/home/lab1006/hashset1.txt offset 100-199
68,c789c84006c3f1b15ce33b8e50c83001,/home/lab1006/hashset1.txt offset 200-267
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 20 hashset.txt
%%%% HASHDEEP-1.0
%%%% size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 -p 20 hashset.txt
##
20,f9892b0092d2fc81fa7b50d6ad6f85a2,/home/lab1006/hashset.txt offset 0-19
20,47578064338e85b10b8f53aa72a62e89,/home/lab1006/hashset.txt offset 20-39
18,007605755622844bc154e50b240f6e20,/home/lab1006/hashset.txt offset 40-57
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

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

Roll no. : 53
Name: Shreya Kamath
Date: 17th August, 2023.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset5.txt *
/home/lab1006/Desktop: Is a directory
/home/lab1006/Documents: Is a directory
/home/lab1006/Downloads: Is a directory
/home/lab1006/file2.txt
/home/lab1006/hashset1.txt
/home/lab1006/hashoutput.txt
/home/lab1006/mojo: Is a directory
/home/lab1006/Music: Is a directory
/home/lab1006/hashtext1.txt
/home/lab1006/hashset.txt
/home/lab1006/Pictures: Is a directory
/home/lab1006/hashset4.txt
/home/lab1006/Public: Is a directory
/home/lab1006/T13-53-CNS: Is a directory
/home/lab1006/Templates: Is a directory
/home/lab1006/Videos: Is a directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m -s hashset5.txt *
md5deep: -s: No such file or directory
md5deep: Unable to load any matching files.
Try `md5deep -h` for more information.
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5 -s -m hashset5.txt *

Command 'md5' not found, did you mean:

  command 'mdl' from snap mdl (0.11.0)
  command 'cd5' from deb cd5
  command 'mdu' from deb mtools
  command 'mdp' from deb mdp

See 'snap info <snapname>' for additional versions.

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -m hashset5.txt *
/home/lab1006/file2.txt
/home/lab1006/hashoutput.txt
/home/lab1006/hashset.txt
/home/lab1006/hashset4.txt
/home/lab1006/hashtext1.txt
/home/lab1006/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x newhashset.txt *
/home/lab1006/file3inverse
/home/lab1006/newhashset.txt
/home/lab1006/examples.desktop
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

CONCLUSION:

Hence, I have successfully conducted an extensive exploration of forensic auditing tools, particularly Hashdeep. Through these activities, I've gained a deeper understanding of the significance of maintaining data integrity and the role Hashdeep plays in ensuring the authenticity and unaltered state of files, thus reinforcing my comprehension of forensic analysis and its relevance in preserving data reliability.

Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.

LAB ASSIGNMENT NO. 6

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

LAB OUTCOME ATTAINED:

LO 3: Explore the different network reconnaissance tools to gather information about networks.

THEORY:

1. whois

The "whois" command is used to retrieve registration and ownership details of domain names, IP addresses, or ASNs by querying WHOIS databases.
For example: `whois example.com` retrieves information about the domain "example.com".
Attackers use the "whois" command to gather domain ownership, contact details, and registration dates. This information aids in social engineering, spear phishing, and domain hijacking attacks, exploiting vulnerabilities based on the revealed organisational structure and registration history.

2. dig

The "dig" command is a network tool used to perform DNS queries, providing information about domain names, IP addresses, and DNS records. It assists in troubleshooting network issues and verifying DNS configurations.
Options:

- `dig example.com MX` - Retrieves Mail Exchange records for "example.com."
- `dig -x 8.8.8.8` - Performs reverse DNS lookup for IP address 8.8.8.8.
- `dig +short example.com` - Shows only IP addresses associated with "example.com."
- ’dig example.com NS +trace` - Traces delegation path and queries authoritative nameservers for "example.com."
- `dig example.com AAAA +dnssec` - Requests IPv6 addresses with DNSSEC information.
- `dig example.com SOA +noall +answer` - Retrieves Start of Authority record, displaying only the answer section.

3. traceroute

The "traceroute" command is a network diagnostic tool that traces the route and measures the round-trip time of packets as they travel through routers between a source and a destination IP address. It helps identify network paths and potential bottlenecks.
The "traceroute" command works by sending packets with gradually increasing Time-to-Live (TTL) values. As each packet travels through routers, the TTL decreases. When the TTL becomes zero, the router discards the packet and sends an ICMP Time Exceeded message back to the source. By analysing the series of ICMP messages and their round-trip times, "traceroute" maps the network path from the source to the destination. The source IP and port remain constant, while the destination port and TTL change for each packet to build the path and calculate latency.

Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.

4. **Nslookup**

The "nslookup" command is a network utility used to query DNS servers for domain name resolution, IP address retrieval, and DNS record information. It assists in diagnosing DNS-related issues and providing essential network information.

5. **Nikto:**

Nikto is built on LibWhisker (by RFP) and can run on any platform which has a Perl environment. It supports SSL, proxies, host authentication, IDS evasion and more. It can be updated automatically from the command-line, and supports the optional submission of updated version data back to the maintainers.
Generally, vulnerabilities in websites can lead to various attacks such as Cross-Site Scripting (XSS), SQL Injection, Remote Code Execution, and Information Disclosure. The potential impact of an exploit depends on the nature of the vulnerability and the attacker's intentions, which could include data theft, website defacement, unauthorised access, and more. Always prioritise security patching and follow best practices to mitigate such risks.

6. **Dmitry:**

DMitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux command line application with the ability to gather as much information as possible about a host.
Basic functionality of DMitry allows for information to be gathered about a target host from a simple whois lookup on the target to uptime reports and TCP port scans.
The application is considered a tool to assist in information gathering when information is required quickly by removing the need to enter multiple commands and the timely process of searching through data from multiple sources.

1. WHOIS Lookup:
   dmitry -w example.com

2. IP WHOIS Lookup:
   dmitry -wi 8.8.8.8

3. Retrieve Netcraft Info:
   dmitry -n example.com

4. Search for Subdomains:
   dmitry -s example.com

5. Search for Email Addresses:
   dmitry -e example.com

6. TCP Port Scan:
   dmitry -p example.com

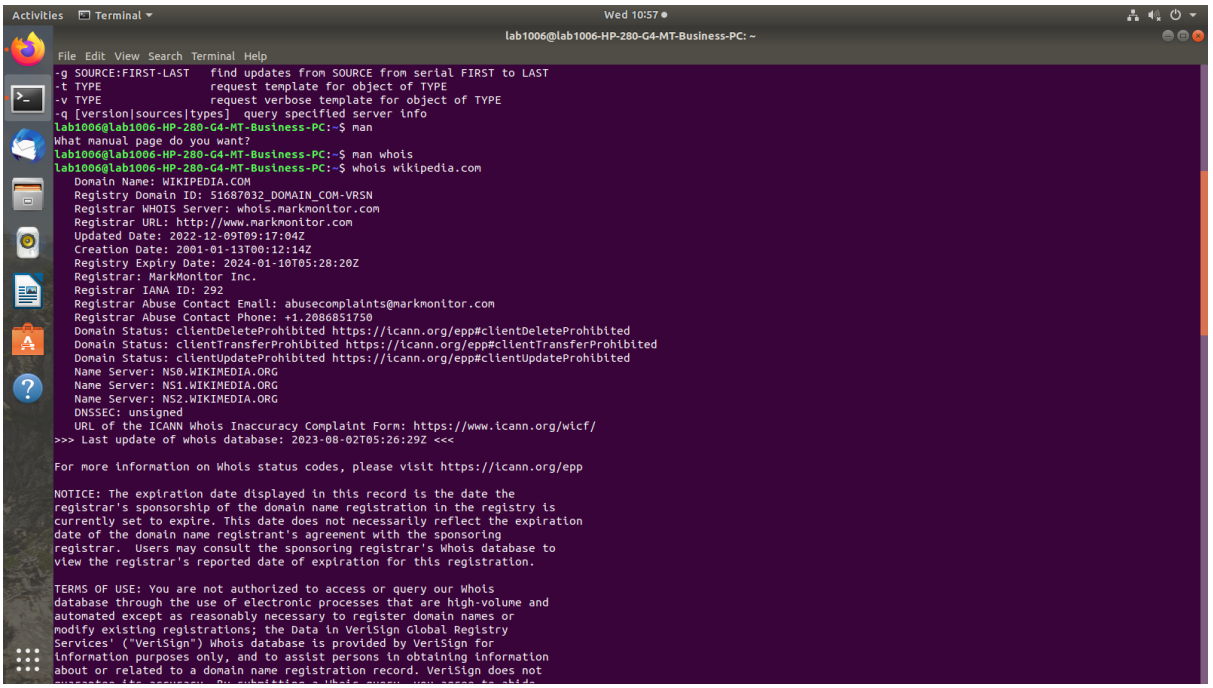
7. Save Output to example.txt:
   dmitry -s -e -p example.com > example.txt

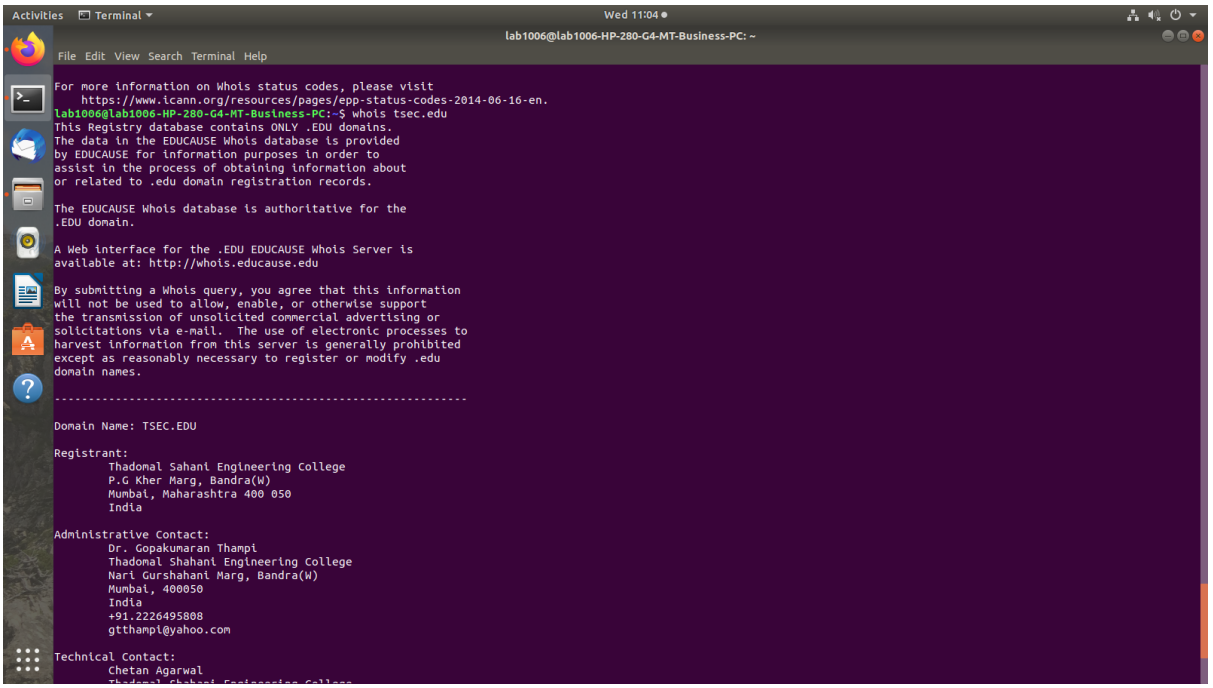
Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.

Email Harvesting Command:
dmitry -e example.com

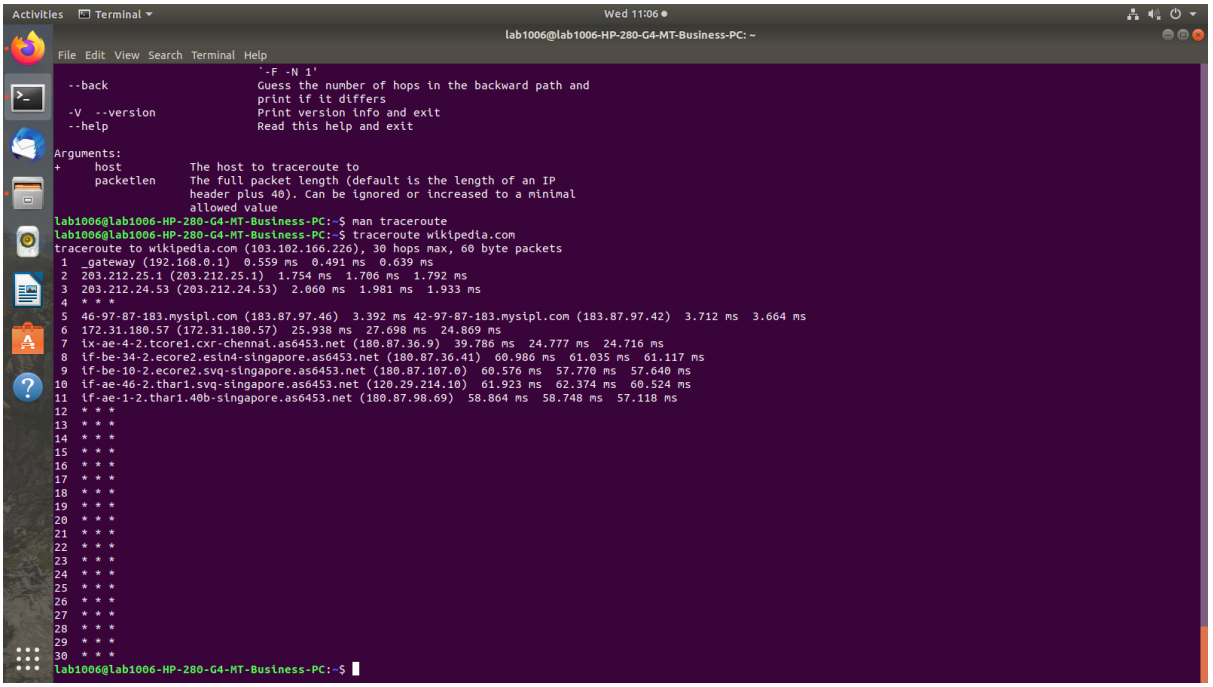
Subdomain Harvesting Command:
dmitry -s example.com

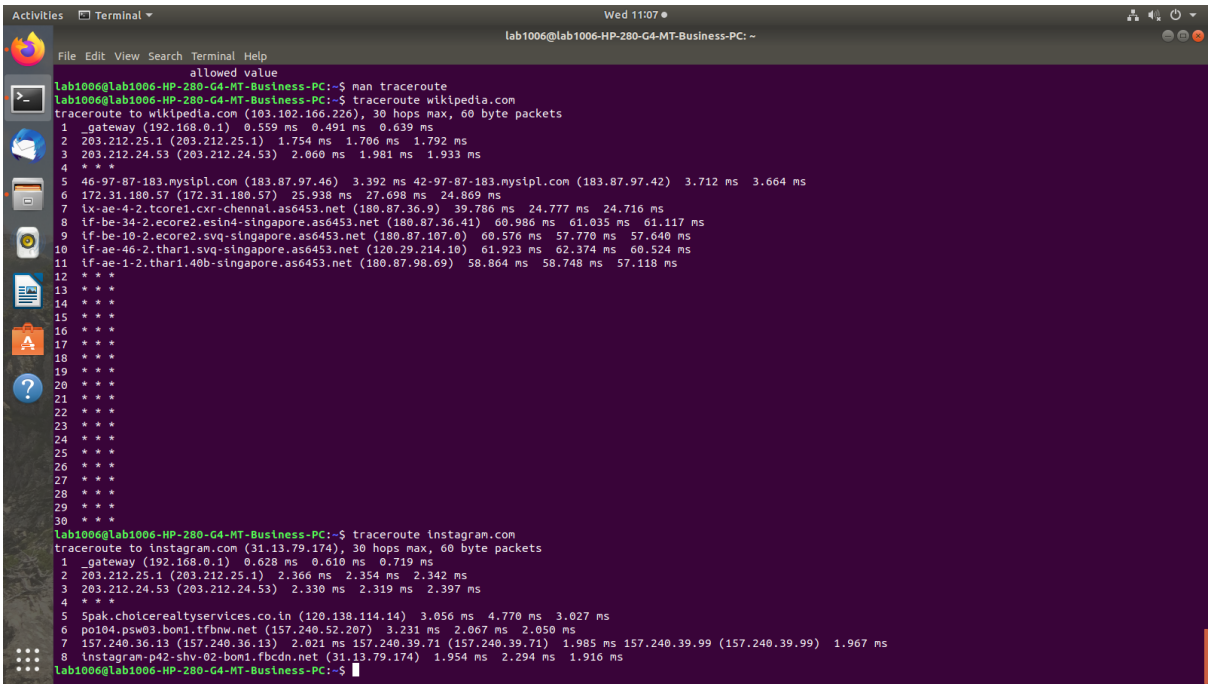
OUTPUT:



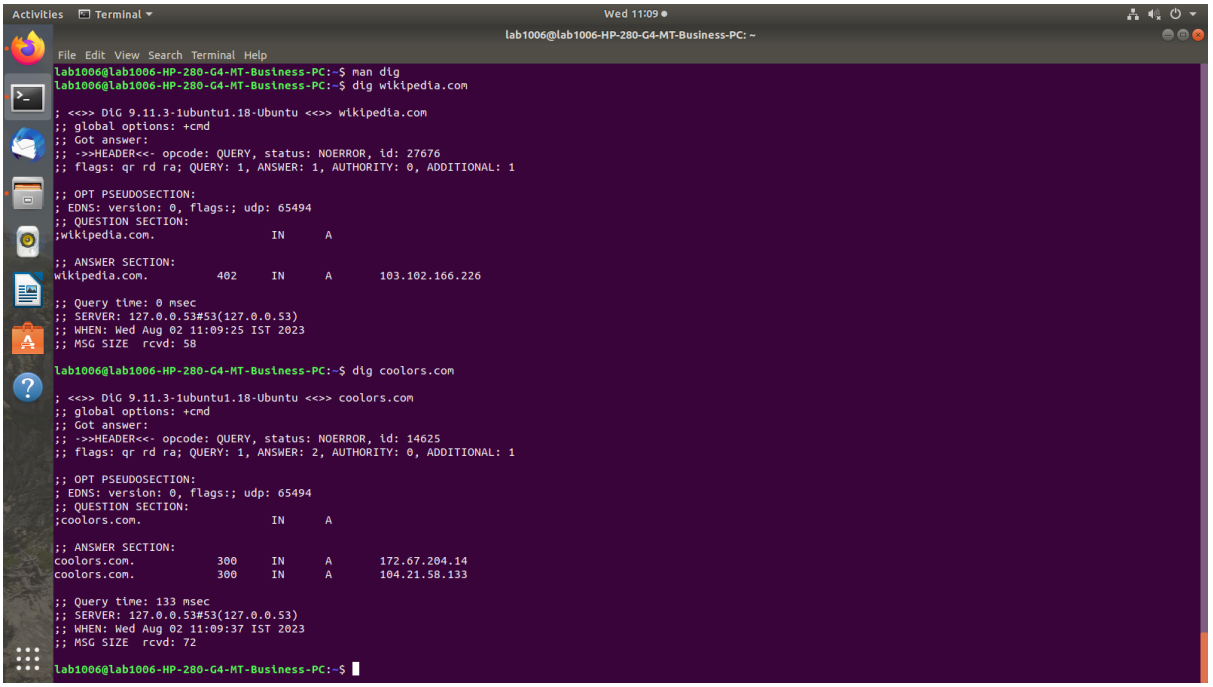


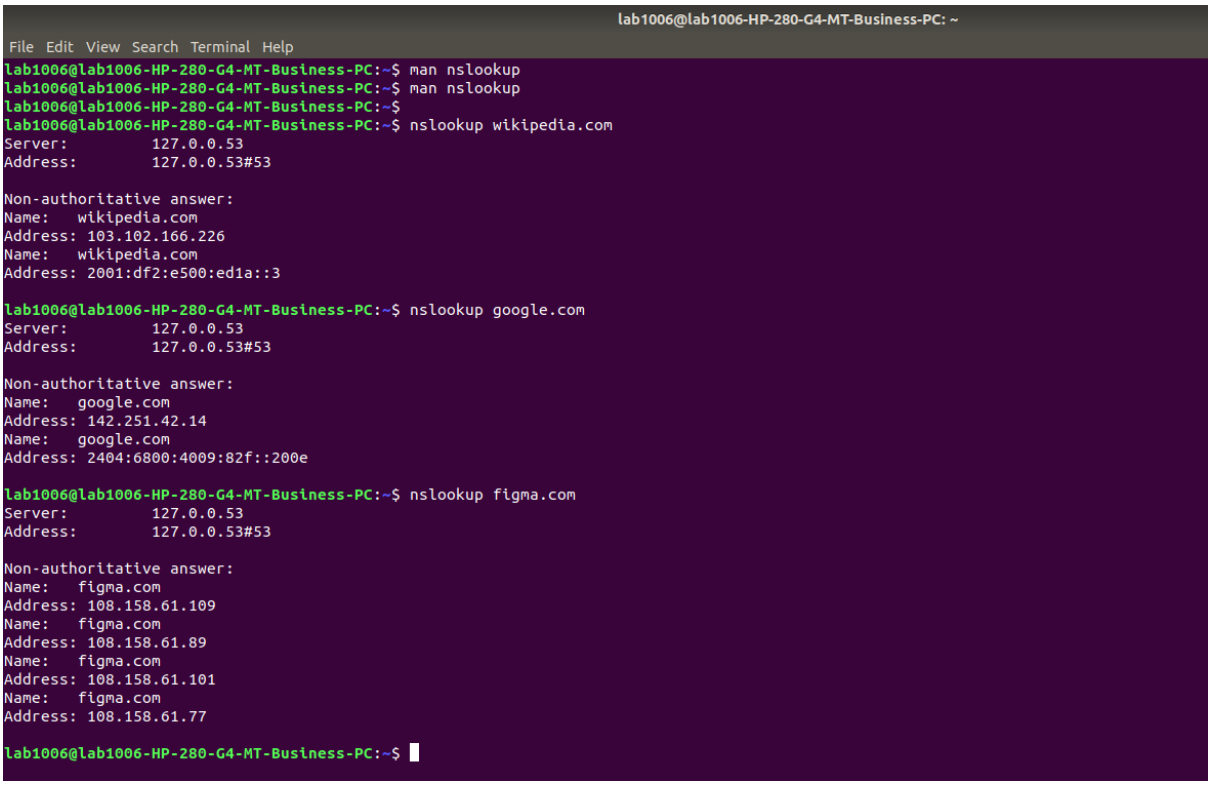
Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.





Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.



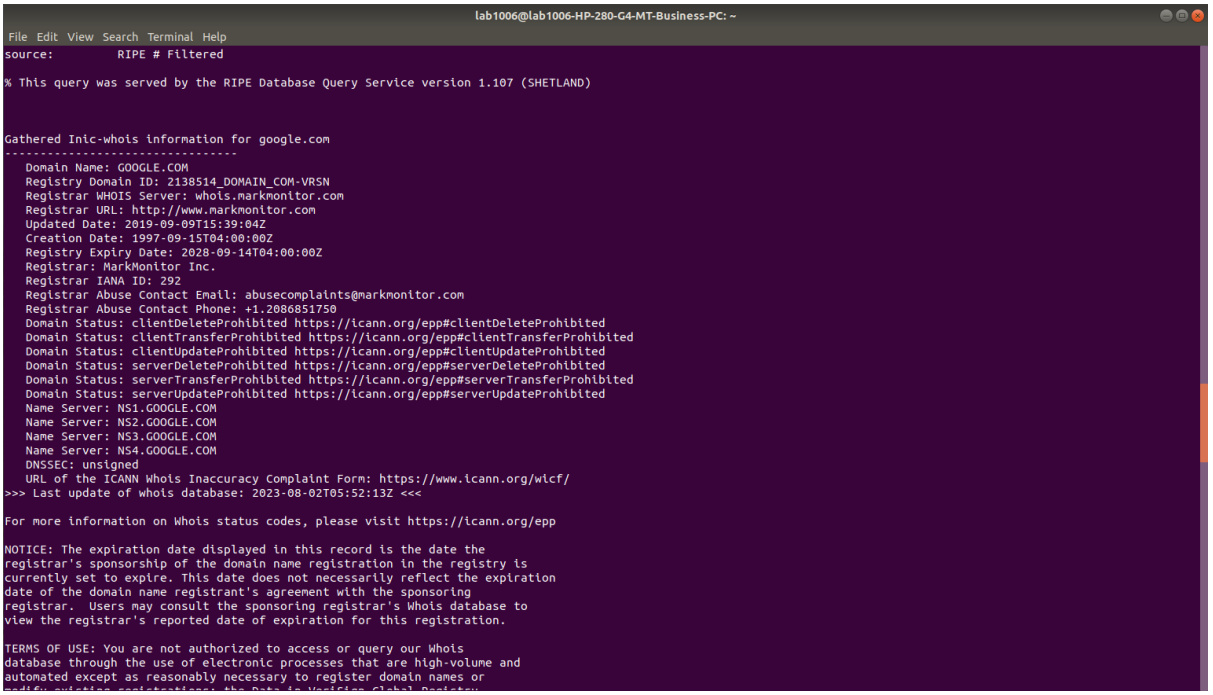


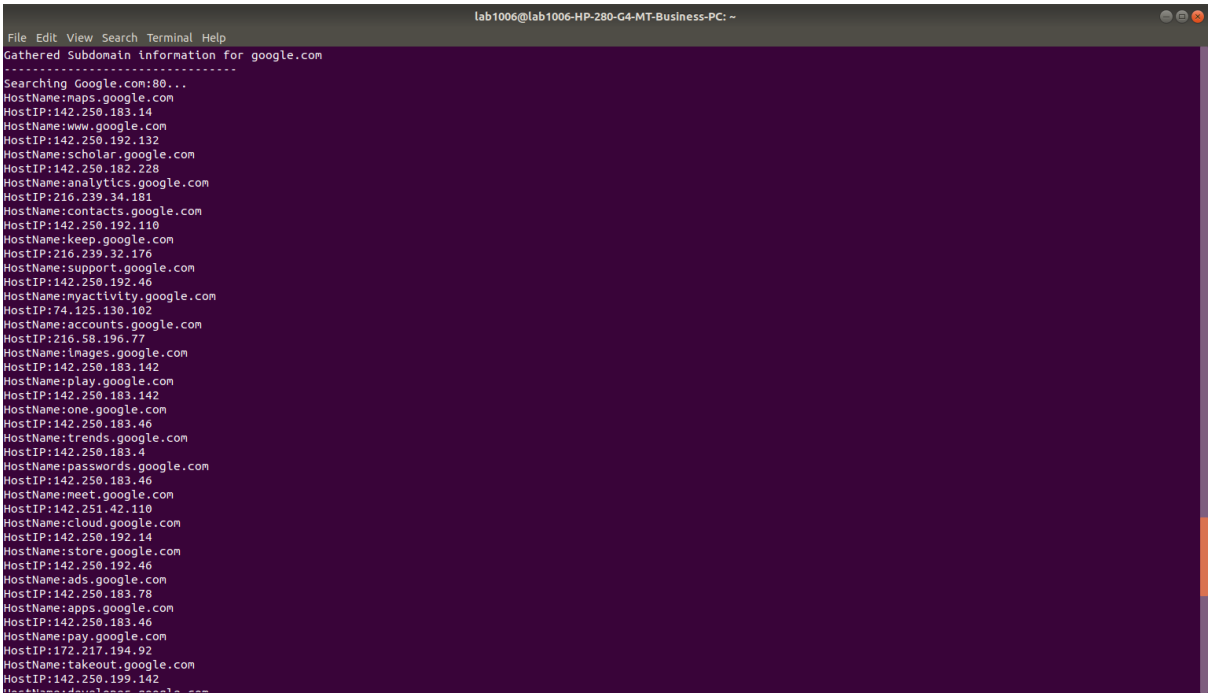
Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h wikipedia.com
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          103.102.166.226
+ Target Hostname:    wikipedia.com
+ Target Port:        80
+ Start Time:         2023-08-02 11:16:43 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.18.0
+ The anti-clickjacking X-Frame-Options header is not present.
+ Root page / redirects to: https://wikipedia.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)

^Clab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h tsec.edu
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          162.241.70.62
+ Target Hostname:    tsec.edu
+ Target Port:        80
+ Start Time:         2023-08-02 11:19:35 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Root page / redirects to: https://tsec.edu/
^Clab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

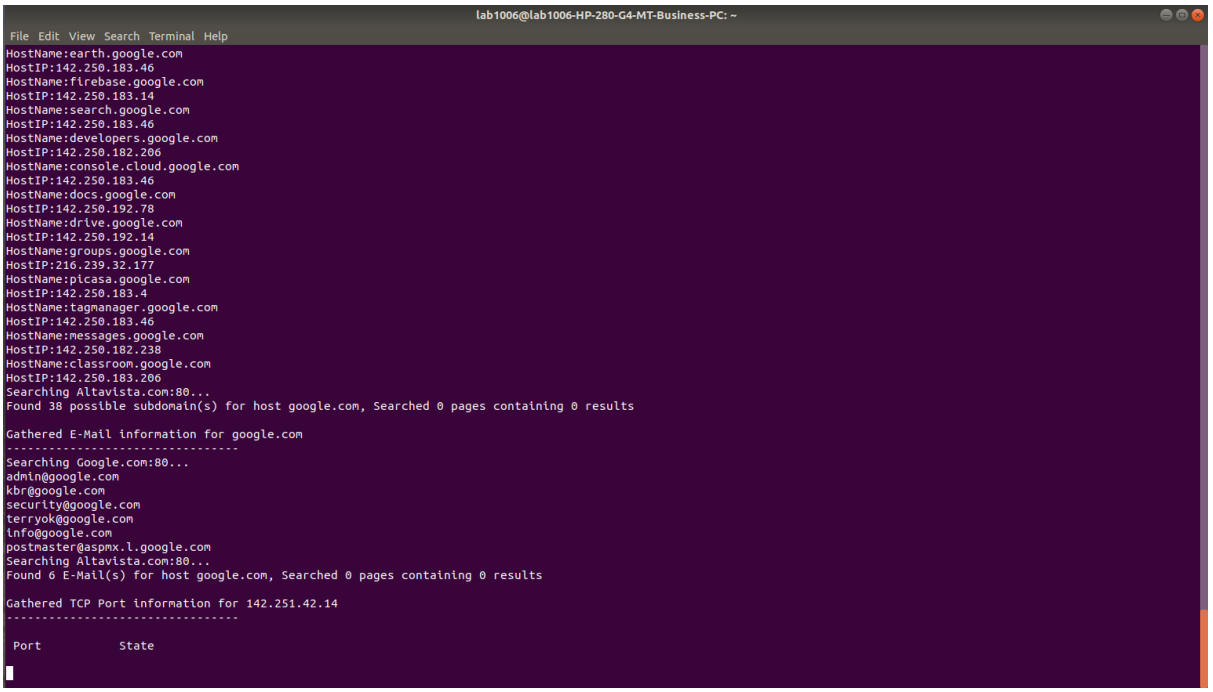
```
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File  Edit  View  Search  Terminal  Help
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Root page / redirects to: https://tsec.edu/
^Clab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man dmitry
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dmitry google.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:142.251.42.14
HostName:google.com

Gathered Inet-whois information for 142.251.42.14
---------------------------------

inetnum:        142.248.0.0 - 143.46.255.255
netname:        NON-RIPE-NCC-MANAGED-ADDRESS-BLOCK
descr:          IPv4 address block not managed by the RIPE NCC
remarks:        -----------------------------------------------------
remarks:
remarks:        For registration information,
remarks:        you can consult the following sources:
remarks:
remarks:        IANA
remarks:        http://www.iana.org/assignments/ipv4-address-space
remarks:        http://www.iana.org/assignments/iana-ipv4-special-registry
remarks:        http://www.iana.org/assignments/ipv4-recovered-address-space
remarks:
remarks:        AFRINIC (Africa)
remarks:        http://www.afrinic.net/ whois.afrinic.net
remarks:
remarks:        APNIC (Asia Pacific)
remarks:        http://www.apnic.net/ whois.apnic.net
remarks:
remarks:        ARIN (Northern America)
remarks:        http://www.arin.net/ whois.arin.net
remarks:
remarks:        LACNIC (Latin America and the Carribean)
remarks:        http://www.lacnic.net/ whois.lacnic.net
remarks:
remarks:        -----------------------------------------------------
country:        EU # Country is really world wide
admin-c:        IANA1-RIPE
tech-c:         IANA1-RIPE
status:         ALLOCATED UNSPECIFIED
mnt-by:         RIPE-NCC-HM-MNT
```

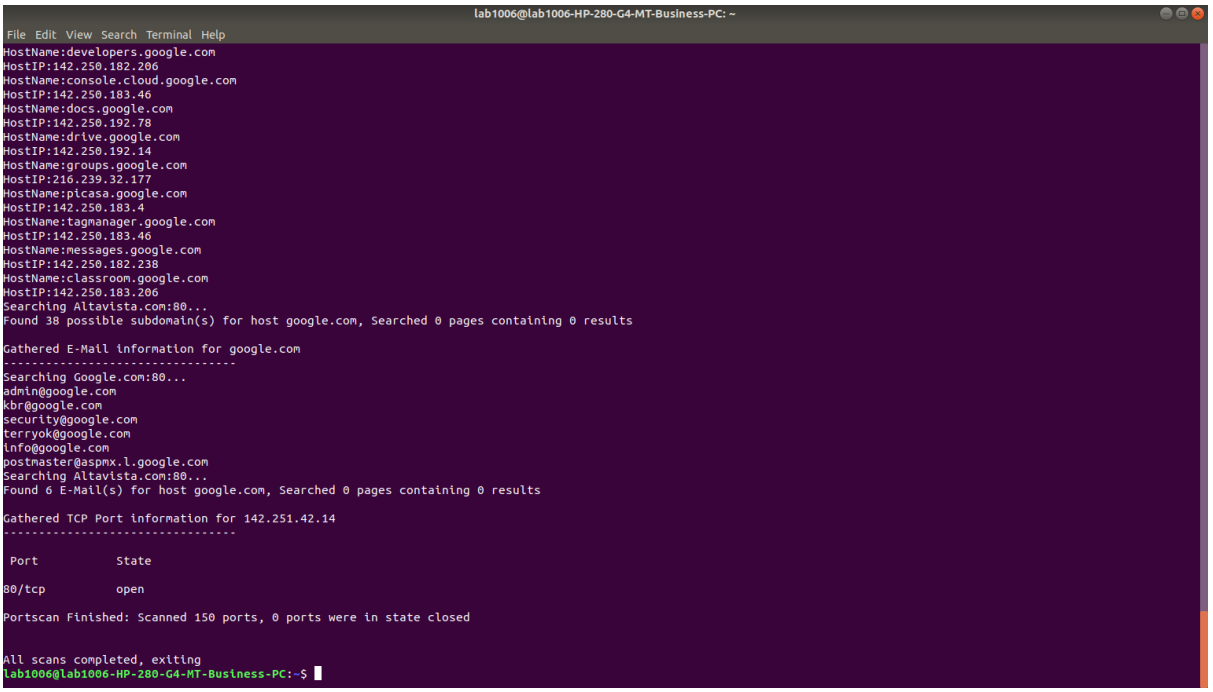
Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.





Roll no. : 53
Name: Shreya Kamath
Date: 7th August, 2023.





CONCLUSION:

Hence, I have successfully executed a comprehensive study of network reconnaissance tools, including WHOIS, dig, traceroute, nslookup, Nikto, and Dmitry. These tools revealed invaluable data about network configurations, domain ownership, and potential vulnerabilities. This practical exposure enhances my understanding of network analysis, security assessment, and the critical role these tools play in ensuring robust network security.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

LAB ASSIGNMENT NO. 7

AIM: Study of packet sniffer tools Wireshark and TCPDUMP.

LAB OUTCOME ATTAINED:

LO 3: Explore the different network reconnaissance tools to gather information about networks.

THEORY:

TCPdump is a widely used network packet analyzer command-line tool. It allows users to capture and analyse network traffic on a system. It's particularly valuable for diagnosing network issues, troubleshooting, and monitoring network activities.

Installation of TCPdump:
sudo apt-get install tcpdump

Choosing an interface:
By default, tcpdump captures packets on all interfaces. To view a summary of available interfaces, run the command:
tcpdump -D

Basic command for sniffing
 # tcpdump -n
The -n parameter is given to stop tcpdump from resolving ip addresses to hostnames.
The verbose switch comes in handy to increase the display resolution of this packet. Here is a quick example:
tcpdump -v -n

Selecting packets with specific protocol
 # tcpdump -n tcp
#tcpdump -n icmp

Capturing traffic from particular host or port
Expressions can be used to specify source ip, destination ip, and port numbers. The next example picks up all those packets with source address 172.16.92.1
 # tcpdump -n src 172.16.92.1
 # tcpdump -n dst 172.16.92.1
 # tcpdump -n port 80
 # tcpdump port 80

Specific Packets from specific port
 # tcpdump udp and src port 53

Observing packets within a specific port range
 # tcpdump -n portrange 1-80

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

It shows all packets whose source or destination port is between 1 to 80
tcpdump -n src port 443

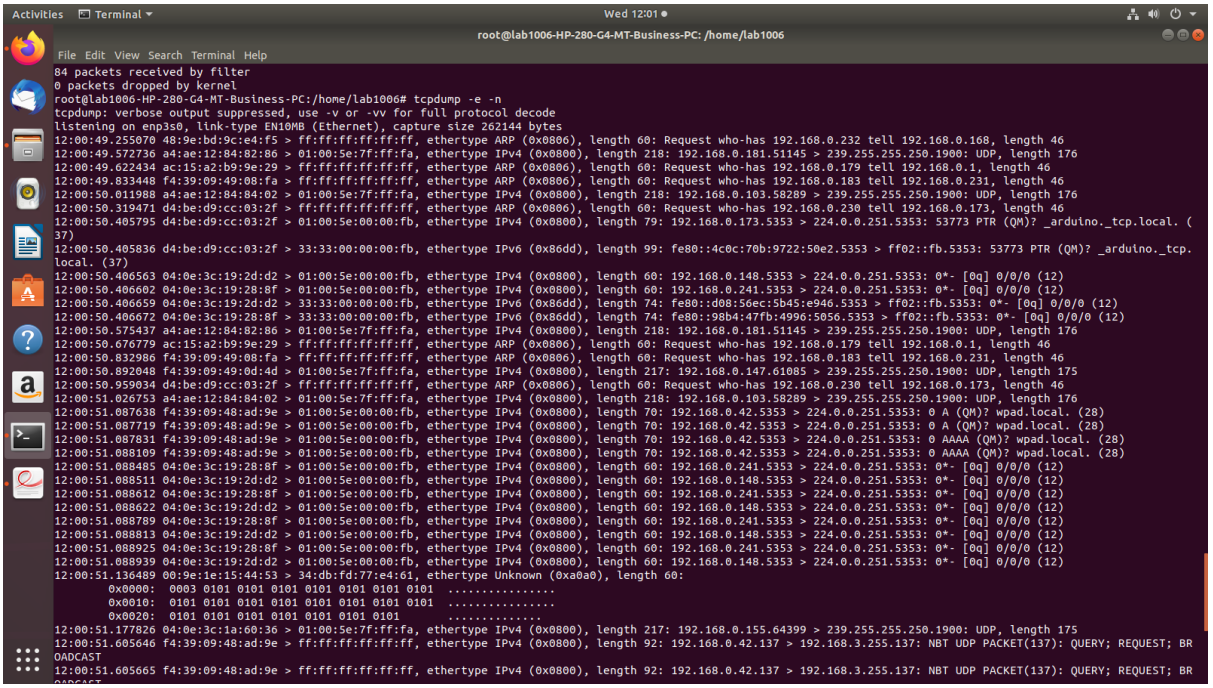
OUTPUT:

```
(socket: Operation not permitted)
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo su
[sudo] password for lab1006:
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:56:26.322051 ARP, Request who-has 192.168.0.183 tell 192.168.0.212, length 46
11:56:26.363690 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74, length 300
11:56:26.546788 IP 192.168.0.234.54309 > 239.255.255.250.1900: UDP, length 176
11:56:26.807417 IP 192.168.0.249.137 > 192.168.0.255.137: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
11:56:26.868992 ARP, Request who-has 192.168.0.183 tell 192.168.0.212, length 46
11:56:26.958281 ARP, Request who-has 192.168.0.227 tell 192.168.0.125, length 46
11:56:27.547501 IP 192.168.0.242.60818 > 239.255.255.250.1900: UDP, length 175
11:56:27.558155 IP 192.168.0.234.54309 > 239.255.255.250.1900: UDP, length 176
11:56:27.708811 ARP, Request who-has 192.168.0.227 tell 192.168.0.125, length 46
11:56:27.745038 IP 192.168.0.199.5353 > 224.0.0.251.5353: 0 PTR (QU)? _microsoft_mcc._tcp.local. (43)
11:56:27.745081 IP6 fe80::e8f8:c12e:dd5e:4599.5353 > ff02::fb.5353: 0 PTR (QU)? _microsoft_mcc._tcp.local. (43)
11:56:27.745201 IP 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:27.745296 IP6 fe80::98b4:47fb:4996:5056.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:27.745655 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:27.745885 IP6 fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:27.868763 ARP, Request who-has 192.168.0.183 tell 192.168.0.212, length 46
11:56:27.872449 IP6 fe80::3011:4165:bb89:8983 > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
11:56:28.044266 IP6 fe80::3011:4165:bb89:8983 > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
11:56:28.054712 IP 192.168.0.167.55976 > 239.255.255.250.1900: UDP, length 176
11:56:28.556738 IP 192.168.0.242.60818 > 239.255.255.250.1900: UDP, length 175
11:56:28.708674 ARP, Request who-has 192.168.0.227 tell 192.168.0.125, length 46
11:56:28.744389 IP 192.168.0.199.5353 > 224.0.0.251.5353: 0 PTR (QM)? _microsoft_mcc._tcp.local. (43)
11:56:28.744431 IP6 fe80::e8f8:c12e:dd5e:4599.5353 > ff02::fb.5353: 0 PTR (QM)? _microsoft_mcc._tcp.local. (43)
11:56:28.744488 IP 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:28.744556 IP6 fe80::98b4:47fb:4996:5056.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:28.745095 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:28.745295 IP6 fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:28.759744 IP 192.168.0.173.5353 > 224.0.0.251.5353: 25873 PTR (QM)? _arduino._tcp.local. (37)
11:56:28.759801 IP6 fe80::4c0c:70b:9722:50e2.5353 > ff02::fb.5353: 25873 PTR (QM)? _arduino._tcp.local. (37)
11:56:28.759831 IP 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:28.759939 IP6 fe80::98b4:47fb:4996:5056.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:28.760491 IP 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:56:28.760694 IP6 fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
11:56:29.067397 IP 192.168.0.167.55976 > 239.255.255.250.1900: UDP, length 176
11:56:29.568146 IP 192.168.0.242.60818 > 239.255.255.250.1900: UDP, length 175
11:56:29.933505 IP 192.168.0.212.54376 > 239.255.255.250.1900: UDP, length 176
11:56:30.078769 IP 192.168.0.167.55976 > 239.255.255.250.1900: UDP, length 176
11:56:30.251523 IP 192.168.0.149.62299 > 239.255.255.250.1900: UDP, length 175
11:56:30.577083 IP 192.168.0.242.60818 > 239.255.255.250.1900: UDP, length 175
^C
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -v -n
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:57:54.634904 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.183 tell 192.168.0.129, length 46
11:57:54.645763 IP (tos 0x0, ttl 64, id 2715, offset 0, flags [DF], proto TCP (6), length 427)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], cksum 0xbd20 (correct), seq 2449946025:2449946400, ack 723247426, win 501, options [nop,nop,TS val 426004561
ecr 1696330276], length 375
11:57:54.645894 IP (tos 0x0, ttl 64, id 2716, offset 0, flags [DF], proto TCP (6), length 179)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], cksum 0x9cb8 (correct), seq 375:502, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], lengt
h 127
11:57:54.645947 IP (tos 0x0, ttl 64, id 2717, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [.], cksum 0x725b (correct), seq 502:1902, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], lengt
h 1400
11:57:54.645956 IP (tos 0x0, ttl 64, id 2718, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [.], cksum 0x3813 (correct), seq 1902:3302, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], leng
th 1400
11:57:54.645963 IP (tos 0x0, ttl 64, id 2719, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [.], cksum 0x9c64 (correct), seq 3302:4702, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], leng
th 1400
11:57:54.645969 IP (tos 0x0, ttl 64, id 2720, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [.], cksum 0xaa7b (correct), seq 4702:6102, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], leng
th 1400
11:57:54.645976 IP (tos 0x0, ttl 64, id 2721, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [.], cksum 0x5aad (correct), seq 6102:7502, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], leng
th 1400
11:57:54.646002 IP (tos 0x0, ttl 64, id 2722, offset 0, flags [DF], proto TCP (6), length 115)
    192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], cksum 0xcd62 (correct), seq 7502:7565, ack 1, win 501, options [nop,nop,TS val 426004561 ecr 1696330276], len
gth 63
11:57:54.647845 IP (tos 0x0, ttl 59, id 44107, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xc086 (correct), ack 502, win 2963, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.647881 IP (tos 0x0, ttl 59, id 44108, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xbb0e (correct), ack 1902, win 2963, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.647887 IP (tos 0x0, ttl 59, id 44109, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xb59b (correct), ack 3302, win 2958, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.647895 IP (tos 0x0, ttl 59, id 44106, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xc105 (correct), ack 375, win 2963, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.647898 IP (tos 0x0, ttl 59, id 44110, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xb028 (correct), ack 4702, win 2953, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.647984 IP (tos 0x0, ttl 59, id 44111, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xaab5 (correct), ack 6102, win 2948, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.648228 IP (tos 0x0, ttl 59, id 44112, offset 0, flags [none], proto TCP (6), length 52)
    142.250.183.142.443 > 192.168.0.182.57404: Flags [.], cksum 0xa542 (correct), ack 7502, win 2943, options [nop,nop,TS val 1696340974 ecr 426004561], length 0
11:57:54.648423 IP (tos 0x0, ttl 59, id 44113, offset 0, flags [none], proto TCP (6), length 52)
```

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

```
11:57:55.652394 IP (tos 0x0, ttl 1, id 13855, offset 0, flags [none], proto UDP (17), length 40)
    192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:57:55.668456 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.150 tell 192.168.0.138, length 46
11:57:55.808503 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.226 tell 192.168.0.214, length 46
11:57:55.875091 IP (tos 0x0, ttl 1, id 56459, offset 0, flags [none], proto UDP (17), length 203)
    192.168.0.244.52422 > 239.255.255.250.1900: UDP, length 175
11:57:55.941400 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.179 tell 192.168.0.1, length 46
11:57:56.037519 IP (tos 0x0, ttl 1, id 739, offset 0, flags [none], proto UDP (17), length 204)
    192.168.0.231.63797 > 239.255.255.250.1900: UDP, length 176
11:57:56.041545 IP (tos 0x0, ttl 1, id 25603, offset 0, flags [DF], proto UDP (17), length 201)
    192.168.0.207.52156 > 239.255.255.250.1900: UDP, length 173
11:57:56.141906 IP (tos 0x0, ttl 1, id 44660, offset 0, flags [none], proto UDP (17), length 203)
    192.168.0.175.52366 > 239.255.255.250.1900: UDP, length 175
11:57:56.147798 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.183 tell 192.168.0.129, length 46
11:57:56.228780 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.224 tell 192.168.0.167, length 46
11:57:56.352459 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.224 tell 192.168.0.175, length 46
11:57:56.514920 00:9e:1e:15:44:53 > 34:db:fd:77:e4:61, ethertype Unknown (0xa0a0), length 60:
        0x0000:  0003 0101 0101 0101 0101 0101 0101 0101  ................
        0x0010:  0101 0101 0101 0101 0101 0101 0101 0101  ................
        0x0020:  0101 0101 0101 0101 0101 0101            ............
11:57:56.549537 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.224 tell 192.168.0.101, length 46
11:57:56.587299 IP (tos 0x0, ttl 64, id 56181, offset 0, flags [DF], proto TCP (6), length 98)
    192.168.0.182.40336 > 142.250.67.141.443: Flags [P.], cksum 0x2f62 (correct), seq 144376332:144376378, ack 752583640, win 501, options [nop,nop,TS val 1902569488 ec
r 2327487075], length 46
11:57:56.589288 IP (tos 0x0, ttl 122, id 11714, offset 0, flags [none], proto TCP (6), length 98)
    142.250.67.141.443 > 192.168.0.182.40336: Flags [P.], cksum 0xf7e4 (correct), seq 1:47, ack 46, win 309, options [nop,nop,TS val 2327545079 ecr 1902569488], length
46
11:57:56.589332 IP (tos 0x0, ttl 64, id 56182, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.40336 > 142.250.67.141.443: Flags [.], cksum 0x1cb3 (correct), ack 47, win 501, options [nop,nop,TS val 1902569491 ecr 2327545079], length 0
11:57:56.660732 IP (tos 0x0, ttl 1, id 36916, offset 0, flags [none], proto UDP (17), length 68)
    192.168.0.129.5353 > 224.0.0.251.5353: 0 PTR (QM)? _googlecast._tcp.local. (40)
11:57:56.660771 IP (tos 0x0, ttl 1, id 17773, offset 0, flags [none], proto UDP (17), length 40)
    192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:57:56.661298 IP (tos 0x0, ttl 1, id 13856, offset 0, flags [none], proto UDP (17), length 40)
    192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
11:57:56.729954 IP (tos 0x0, ttl 128, id 19827, offset 0, flags [none], proto UDP (17), length 78)
    192.168.0.168.137 > 192.168.0.255.137: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
11:57:56.897383 IP (tos 0x0, ttl 1, id 9041, offset 0, flags [none], proto UDP (17), length 204)
    192.168.0.190.57147 > 239.255.255.250.1900: UDP, length 176
11:57:56.937331 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.179 tell 192.168.0.1, length 46
^C
84 packets captured
84 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```



Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -e -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:00:49.255070 48:9e:bd:9c:e4:f5 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.232 tell 192.168.0.168, length 46
12:00:49.572736 a4:ae:12:84:82:86 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: 192.168.0.181.51145 > 239.255.255.250.1900: UDP, length 176
12:00:49.622434 ac:15:a2:b9:9e:29 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.179 tell 192.168.0.1, length 46
12:00:49.833448 f4:39:09:49:08:fa > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.183 tell 192.168.0.231, length 46
12:00:50.011988 a4:ae:12:84:84:02 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: 192.168.0.103.58289 > 239.255.255.250.1900: UDP, length 176
12:00:50.319471 d4:be:d9:cc:03:2f > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.230 tell 192.168.0.173, length 46
12:00:50.405795 d4:be:d9:cc:03:2f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 79: 192.168.0.173.5353 > 224.0.0.251.5353: 53773 PTR (QM)? _arduino._tcp.local. (
37)
12:00:50.405836 d4:be:d9:cc:03:2f > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 99: fe80::4c0c:70b:9722:50e2.5353 > ff02::fb.5353: 53773 PTR (QM)? _arduino._tcp.
local. (37)
12:00:50.406563 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:50.406602 04:0e:3c:19:28:8f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:50.406659 04:0e:3c:19:2d:d2 > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 74: fe80::d08:56ec:5b45:e946.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
12:00:50.406672 04:0e:3c:19:28:8f > 33:33:00:00:00:fb, ethertype IPv6 (0x86dd), length 74: fe80::98b4:47fb:4996:5056.5353 > ff02::fb.5353: 0*- [0q] 0/0/0 (12)
12:00:50.575437 a4:ae:12:84:82:86 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: 192.168.0.181.51145 > 239.255.255.250.1900: UDP, length 176
12:00:50.676779 ac:15:a2:b9:9e:29 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.179 tell 192.168.0.1, length 46
12:00:50.832986 f4:39:09:49:08:fa > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.183 tell 192.168.0.231, length 46
12:00:50.892048 f4:39:09:49:0d:4d > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 217: 192.168.0.147.61085 > 239.255.255.250.1900: UDP, length 175
12:00:50.959034 d4:be:d9:cc:03:2f > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.0.230 tell 192.168.0.173, length 46
12:00:51.026753 a4:ae:12:84:84:02 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 218: 192.168.0.103.58289 > 239.255.255.250.1900: UDP, length 176
12:00:51.087638 f4:39:09:48:ad:9e > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 70: 192.168.0.42.5353 > 224.0.0.251.5353: 0 A (QM)? wpad.local. (28)
12:00:51.087719 f4:39:09:48:ad:9e > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 70: 192.168.0.42.5353 > 224.0.0.251.5353: 0 A (QM)? wpad.local. (28)
12:00:51.087831 f4:39:09:48:ad:9e > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 70: 192.168.0.42.5353 > 224.0.0.251.5353: 0 AAAA (QM)? wpad.local. (28)
12:00:51.088109 f4:39:09:48:ad:9e > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 70: 192.168.0.42.5353 > 224.0.0.251.5353: 0 AAAA (QM)? wpad.local. (28)
12:00:51.088485 04:0e:3c:19:28:8f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088511 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088612 04:0e:3c:19:28:8f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088622 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088789 04:0e:3c:19:28:8f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088813 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088925 04:0e:3c:19:28:8f > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.241.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.088939 04:0e:3c:19:2d:d2 > 01:00:5e:00:00:fb, ethertype IPv4 (0x0800), length 60: 192.168.0.148.5353 > 224.0.0.251.5353: 0*- [0q] 0/0/0 (12)
12:00:51.136489 00:9e:1e:15:44:53 > 34:db:fd:77:e4:61, ethertype Unknown (0xa0a0), length 60:
        0x0000:  0003 0101 0101 0101 0101 0101 0101 0101  ................
        0x0010:  0101 0101 0101 0101 0101 0101 0101 0101  ................
        0x0020:  0101 0101 0101 0101 0101 0101 0101       ..............
12:00:51.177826 04:0e:3c:1a:60:36 > 01:00:5e:7f:ff:fa, ethertype IPv4 (0x0800), length 217: 192.168.0.155.64399 > 239.255.255.250.1900: UDP, length 175
12:00:51.605646 f4:39:09:48:ad:9e > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 92: 192.168.0.42.137 > 192.168.3.255.137: NBT UDP PACKET(137): QUERY; REQUEST; BR
OADCAST
12:00:51.605665 f4:39:09:48:ad:9e > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 92: 192.168.0.42.137 > 192.168.3.255.137: NBT UDP PACKET(137): QUERY; REQUEST; BR
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -n src  192.168.0.182
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:02:50.363335 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:02:50.363359 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:02:50.522515 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
12:02:50.538412 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
12:02:52.774977 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 902728857, win 7327, options [nop,nop,TS val 1145898009 ecr 416470929], length 0
12:03:00.311767 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 2450222053:2450222435, ack 723285217, win 501, options [nop,nop,TS val 426310217 ecr 16966
18757], length 382
12:03:00.311803 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 382:1782, ack 1, win 501, options [nop,nop,TS val 426310217 ecr 1696618757], length 1400
12:03:00.311870 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 1782:3182, ack 1, win 501, options [nop,nop,TS val 426310217 ecr 1696618757], length 1400
12:03:00.311889 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 3182:4582, ack 1, win 501, options [nop,nop,TS val 426310217 ecr 1696618757], length 1400
12:03:00.312010 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 4582:5982, ack 1, win 501, options [nop,nop,TS val 426310217 ecr 1696618757], length 1400
12:03:00.312013 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 5982:6075, ack 1, win 501, options [nop,nop,TS val 426310217 ecr 1696618757], length 93
12:03:00.313718 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 0:1400, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313735 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 1400:2800, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313737 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 2800:4200, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313846 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 4200:5600, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313850 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 5600:7000, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313852 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 7000:8400, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1400
12:03:00.313946 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], seq 8400:9632, ack 1, win 7327, options [nop,nop,TS val 1145905548 ecr 416470929], length 1232
12:03:00.438454 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], ack 862, win 501, options [nop,nop,TS val 426310344 ecr 1696646759], length 0
12:03:00.440774 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], ack 1061, win 501, options [nop,nop,TS val 426310346 ecr 1696646762], length 0
12:03:00.440839 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 6075:6121, ack 1061, win 501, options [nop,nop,TS val 426310346 ecr 1696646762], length 46
12:03:00.660391 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 678, win 7327, options [nop,nop,TS val 1145905895 ecr 416478814], length 0
12:03:00.662830 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 716, win 7327, options [nop,nop,TS val 1145905897 ecr 416478817], length 0
12:03:00.662865 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 762, win 7327, options [nop,nop,TS val 1145905897 ecr 416478817], length 0
12:03:00.663201 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], seq 9632:9678, ack 762, win 7327, options [nop,nop,TS val 1145905897 ecr 416478817], length 46
12:03:03.143723 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:03:03.313462 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 6121:6656, ack 1061, win 501, options [nop,nop,TS val 426313219 ecr 1696646763], length 53
5
12:03:03.313499 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 6656:8056, ack 1061, win 501, options [nop,nop,TS val 426313219 ecr 1696646763], length 140
0
12:03:03.313501 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 8056:9456, ack 1061, win 501, options [nop,nop,TS val 426313219 ecr 1696646763], length 140
0
12:03:03.313575 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], seq 9456:10856, ack 1061, win 501, options [nop,nop,TS val 426313219 ecr 1696646763], length 14
00
```

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

```
12:03:03.316865 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 18596:19996, ack 762, win 7327, options [nop,nop,TS val 1145908551 ecr 416478821], length 14
00
12:03:03.317040 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 19996:21396, ack 762, win 7327, options [nop,nop,TS val 1145908551 ecr 416478821], length 14
00
12:03:03.317044 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 21396:22796, ack 762, win 7327, options [nop,nop,TS val 1145908551 ecr 416478821], length 14
00
12:03:03.319238 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 22796:24196, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319242 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 24196:25596, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319243 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 25596:26996, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319246 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 26996:28396, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319287 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 28396:29796, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319289 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 29796:31196, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319290 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 31196:32596, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319377 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 32596:33996, ack 762, win 7327, options [nop,nop,TS val 1145908553 ecr 416481474], length 14
00
12:03:03.319519 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 33996:35396, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319551 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 35396:36796, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319559 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 36796:38196, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319745 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 38196:39596, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319839 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 39596:40996, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319845 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], seq 40996:42396, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 14
00
12:03:03.319961 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], seq 42396:43340, ack 762, win 7327, options [nop,nop,TS val 1145908554 ecr 416481474], length 9
44
12:03:03.456370 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], ack 1982, win 501, options [nop,nop,TS val 426313361 ecr 1696649777], length 0
12:03:03.457974 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], ack 2195, win 501, options [nop,nop,TS val 426313363 ecr 1696649779], length 0
12:03:03.458039 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 28575:28621, ack 2195, win 501, options [nop,nop,TS val 426313363 ecr 1696649779], length
46
^C
71 packets captured
71 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:04:03.668863 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
12:04:03.680528 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -n portrange 1-79
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:06:13.356129 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74, length 300
12:06:20.106423 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74, length 300
12:06:23.051133 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:74, length 300
12:06:23.760201 IP 192.168.0.182.58135 > 192.168.0.1.53: 56866+ [1au] A? docs.google.com. (44)
12:06:23.761539 IP 192.168.0.1.53 > 192.168.0.182.58135: 56866 1/0/1 A 142.250.192.78 (60)
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -n src 192.168.0.182
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:07:36.254761 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:07:36.912758 IP 192.168.0.182.5353 > 224.0.0.251.5353: 0*- [0q] 2/0/0 (Cache flush) AAAA fe80::bbb9:308f:e56e:ecaa, (Cache flush) A 192.168.0.182 (96)
12:07:39.676165 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 902734872, win 7327, options [nop,nop,TS val 1146184901 ecr 416757825], length 0
12:07:44.877925 ARP, Request who-has 192.168.0.1 tell 192.168.0.182, length 28
12:07:51.363081 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 2450260655:2450260701, ack 723289559, win 501, options [nop,nop,TS val 426601258 ecr 16968
84915], length 46
12:07:51.363433 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [P.], seq 46:77, ack 1, win 501, options [nop,nop,TS val 426601259 ecr 1696884915], length 31
12:07:51.363462 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [F.], seq 77, ack 1, win 501, options [nop,nop,TS val 426601259 ecr 1696884915], length 0
12:07:51.365222 IP 192.168.0.182.57404 > 142.250.183.142.443: Flags [.], ack 2, win 501, options [nop,nop,TS val 426601261 ecr 1696937681], length 0
12:07:51.451584 IP 192.168.0.182.5353 > 224.0.0.251.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afpovertcp._tcp.local. PTR (QM)? _smb.
_tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (118)
12:07:53.837761 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:07:56.598092 IP 192.168.0.182.34564 > 142.250.67.141.443: Flags [P.], seq 2022881181:2022881227, ack 2578354394, win 501, options [nop,nop,TS val 1903169479 ecr 3613
841975], length 46
12:07:56.600228 IP 192.168.0.182.34564 > 142.250.67.141.443: Flags [.], ack 47, win 501, options [nop,nop,TS val 1903169481 ecr 3613899976], length 0
12:08:05.277928 IP 192.168.0.182.60034 > 142.250.192.78.443: Flags [.], ack 162, win 7327, options [nop,nop,TS val 1146210502 ecr 416783427], length 0
12:08:06.950958 ARP, Reply 192.168.0.182 is-at 04:0e:3c:1a:60:28, length 28
12:08:07.208331 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
12:08:07.219993 IP 192.168.0.182 > 192.168.0.1: ICMP 192.168.0.182 udp port 137 unreachable, length 86
12:08:20.876322 IP 192.168.0.182.38464 > 192.168.0.1.53: 33614+ [1au] A? connectivity-check.ubuntu.com. (58)
12:08:20.876584 IP 192.168.0.182.54241 > 192.168.0.1.53: 24423+ [1au] AAAA? connectivity-check.ubuntu.com. (58)
^C
18 packets captured
18 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

```
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# tcpdump -nnvvS src 192.168.0.182 and dst port 443
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:13:38.648451 IP (tos 0x0, ttl 64, id 8700, offset 0, flags [DF], proto TCP (6), length 98)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [P.], cksum 0x9392 (correct), seq 1118152770:1118152816, ack 401697757, win 501, options [nop,nop,TS val 1443630945
 ecr 2073072557], length 46
12:13:38.650313 IP (tos 0x0, ttl 64, id 8701, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [.], cksum 0xe798 (correct), seq 1118152816, ack 401697803, win 501, options [nop,nop,TS val 1443630947 ecr 2073130
561], length 0
12:13:41.997806 IP (tos 0x0, ttl 64, id 23323, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0x30e8 (correct), seq 3415550621, ack 4159023669, win 501, options [nop,nop,TS val 4105395287 ecr 23888370
24], length 0
12:13:42.223861 IP (tos 0x0, ttl 64, id 60338, offset 0, flags [DF], proto TCP (6), length 95)
    192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], cksum 0xe6c0 (correct), seq 2221851728:2221851771, ack 902743322, win 7327, options [nop,nop,TS val 1146547455
 ecr 417114454], length 43
12:13:42.507644 IP (tos 0x0, ttl 64, id 23324, offset 0, flags [DF], proto TCP (6), length 495)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [P.], cksum 0x4e2f (correct), seq 3415550622:3415551065, ack 4159023669, win 501, options [nop,nop,TS val 4105395797
ecr 2388847264], length 443
12:13:42.606177 IP (tos 0x0, ttl 64, id 23325, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0x0068 (correct), seq 3415551065, ack 4159024281, win 501, options [nop,nop,TS val 4105395895 ecr 23888477
76], length 0
12:13:49.746703 IP (tos 0x0, ttl 64, id 23326, offset 0, flags [DF], proto TCP (6), length 495)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [P.], cksum 0x914e (correct), seq 3415551065:3415551508, ack 4159024281, win 501, options [nop,nop,TS val 4105403036
ecr 2388847776], length 443
12:13:49.845135 IP (tos 0x0, ttl 64, id 23327, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0xc3ba (correct), seq 3415551508, ack 4159024893, win 501, options [nop,nop,TS val 4105403134 ecr 23888550
15], length 0
12:13:58.867388 IP (tos 0x0, ttl 64, id 55118, offset 0, flags [DF], proto TCP (6), length 98)
    192.168.0.182.54162 > 142.250.183.142.443: Flags [P.], cksum 0x044e (correct), seq 3780532617:3780532663, ack 1856904268, win 501, options [nop,nop,TS val 426968770
 ecr 3504007057], length 46
12:13:58.913806 IP (tos 0x0, ttl 64, id 55119, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.54162 > 142.250.183.142.443: Flags [.], cksum 0x14c7 (correct), seq 3780532663, ack 1856904314, win 501, options [nop,nop,TS val 426968816 ecr 3504065
366], length 0
12:13:59.917760 IP (tos 0x0, ttl 64, id 23328, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0x9c62 (correct), seq 3415551507, ack 4159024893, win 501, options [nop,nop,TS val 4105413207 ecr 23888550
15], length 0
12:14:01.306581 IP (tos 0x0, ttl 64, id 55120, offset 0, flags [DF], proto TCP (6), length 425)
    192.168.0.182.54162 > 142.250.183.142.443: Flags [P.], cksum 0xeaec (correct), seq 3780532663:3780533036, ack 1856904314, win 501, options [nop,nop,TS val 426971209
 ecr 3504065366], length 373
12:14:01.306644 IP (tos 0x0, ttl 64, id 55121, offset 0, flags [DF], proto TCP (6), length 1452)
    192.168.0.182.54162 > 142.250.183.142.443: Flags [.], cksum 0x61ca (correct), seq 3780533036:3780534436, ack 1856904314, win 501, options [nop,nop,TS val 426971209
ecr 3504065366], length 1400
```

```
0412], length 0
12:14:14.029133 IP (tos 0x0, ttl 64, id 59269, offset 0, flags [DF], proto TCP (6), length 90)
    192.168.0.182.43934 > 103.102.166.224.443: Flags [P.], cksum 0x1150 (correct), seq 3777618545:3777618583, ack 3182039794, win 501, options [nop,nop,TS val 148497424
6 ecr 1808780412], length 38
12:14:16.135515 IP (tos 0x0, ttl 64, id 60351, offset 0, flags [DF], proto TCP (6), length 95)
    192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], cksum 0x509e (correct), seq 2221860918:2221860961, ack 902744258, win 7327, options [nop,nop,TS val 1146581367
 ecr 417142086], length 43
12:14:20.909604 IP (tos 0x0, ttl 64, id 23332, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0xf4a8 (correct), seq 3415551950, ack 4159025505, win 501, options [nop,nop,TS val 4105434199 ecr 23888759
05], length 0
12:14:27.747550 IP (tos 0x0, ttl 64, id 60352, offset 0, flags [DF], proto TCP (6), length 95)
    192.168.0.182.60034 > 142.250.192.78.443: Flags [P.], cksum 0x9274 (correct), seq 2221860961:2221861004, ack 902744258, win 7327, options [nop,nop,TS val 1146592979
 ecr 417154301], length 43
12:14:28.080112 IP (tos 0x0, ttl 64, id 23333, offset 0, flags [DF], proto TCP (6), length 495)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [P.], cksum 0xb938 (correct), seq 3415551951:3415552394, ack 4159025505, win 501, options [nop,nop,TS val 4105441370
ecr 2388886177], length 443
12:14:28.178586 IP (tos 0x0, ttl 64, id 23334, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0x8ffe (correct), seq 3415552394, ack 4159026117, win 501, options [nop,nop,TS val 4105441468 ecr 23888933
50], length 0
12:14:28.735516 IP (tos 0x0, ttl 64, id 60353, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.60034 > 142.250.192.78.443: Flags [.], cksum 0x7b58 (correct), seq 2221861004, ack 902744319, win 7327, options [nop,nop,TS val 1146593967 ecr 4171668
99], length 0
12:14:32.978136 IP (tos 0x0, ttl 64, id 8702, offset 0, flags [DF], proto TCP (6), length 98)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [P.], cksum 0x9192 (correct), seq 1118152816:1118152862, ack 401697803, win 501, options [nop,nop,TS val 1443685276
 ecr 2073130561], length 46
12:14:32.978341 IP (tos 0x0, ttl 64, id 8703, offset 0, flags [DF], proto TCP (6), length 83)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [P.], cksum 0x91ab (correct), seq 1118152862:1118152893, ack 401697803, win 501, options [nop,nop,TS val 1443685276
 ecr 2073130561], length 31
12:14:32.978358 IP (tos 0x0, ttl 64, id 8704, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [F.], cksum 0x1311 (correct), seq 1118152893, ack 401697803, win 501, options [nop,nop,TS val 1443685276 ecr 207313
0561], length 0
12:14:32.980650 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.46146 > 142.250.192.132.443: Flags [.], cksum 0x3ed0 (correct), seq 1118152894, ack 401697804, win 501, options [nop,nop,TS val 1443685279 ecr 2073184
893], length 0
12:14:35.475572 IP (tos 0x0, ttl 64, id 23335, offset 0, flags [DF], proto TCP (6), length 495)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [P.], cksum 0x818f (correct), seq 3415552394:3415552837, ack 4159026117, win 501, options [nop,nop,TS val 4105448766
ecr 2388893350], length 443
12:14:35.575270 IP (tos 0x0, ttl 64, id 23336, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.182.41056 > 54.192.111.73.443: Flags [.], cksum 0x5217 (correct), seq 3415552837, ack 4159026729, win 501, options [nop,nop,TS val 4105448865 ecr 23889007
45], length 0
^C
58 packets captured
58 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

CONCLUSION:

Hence, I have learnt the fundamentals of Wireshark and sniffing tool tcpdump and executed commands to capture packets in different ways.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

LAB 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, etc.

LAB OUTCOME ATTAINED:

LO 3: Explore the different network reconnaissance tools to gather information about networks.

THEORY:

While Nmap has grown in functionality over the years, it began as an efficient port scanner, and that remains its core function. The simple command **nmap *<target>*** scans 1,000 TCP ports on the host *<target>*. While many port scanners have traditionally lumped all ports into the open or closed states, Nmap is much more granular. It divides ports into six states: open, closed, filtered, unfiltered, open|filtered, or closed|filtered. These states are not intrinsic properties of the port itself, but describe how Nmap sees them. For example, an Nmap scan from the same network as the target may show port 135/tcp as open, while a scan at the same time with the same options from across the Internet might show that port as filtered.

The six port states recognized by Nmap:

1. Open

An application is actively accepting TCP connections, UDP datagrams or SCTP associations on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users. Open ports are also interesting for non-security scans because they show services available for use on the network.

2. closed

A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up. Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next.

3. Filtered

Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common. This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically.

4. unfiltered

The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

5. open|filtered

Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response. The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way.

6. closed|filtered

This state is used when Nmap is unable to determine whether a port is closed or filtered. It is only used for the IP ID idle scan.

COMMANDS FOR VARIOUS PORT SCANNING TECHNIQUES:

1. TCP Connect Scan:
   - Command: `*nmap -sT target_ip*`
   - Explanation: TCP Connect Scan is a basic port scanning technique. It establishes a full connection to each target port to determine whether it's open or closed. Open ports will complete the connection, while closed ports will result in a refusal.

2. TCP SYN Scan:
   - Command: `*nmap -sS target_ip*`
   - Explanation: TCP SYN Scan sends SYN packets to target ports and analyses the response. If a SYN/ACK is received, the port is open; if a RST is received, it's closed. This scan is stealthy and doesn't complete the full connection.

3. FIN Scan:
   - Command: `*nmap -sF target_ip*`
   - Explanation: FIN Scan sends FIN packets to target ports. If the port is closed, it should respond with a RST. However, if the port is open, it may ignore the packet. This scan can be used to identify firewall filtering.

4. Null Scan:
   - Command: `*nmap -sN target_ip*`
   - Explanation: Null Scan sends packets with no flags set to target ports. If the port is closed, it should respond with a RST. If it's open, it may ignore the packet. Similar to the FIN Scan, it can identify firewall filtering.

5. XMAS Scan:
   - Command: `*nmap -sX target_ip*`
   - Explanation: XMAS Scan sends packets with multiple TCP flags set (FIN, URG, and PSH) to target ports. Similar to Null and FIN Scans, it's used to identify filtering or state of the ports.

6. ACK Scan:

- Command: `*nmap -sA target_ip*`
- Explanation: ACK Scan sends packets with only the ACK flag set to target ports. It's used to detect stateful filtering systems. If it receives an RST, the port is unfiltered.

7. Ping Sweep:
- Command: `*nmap -sn target_subnet*`
- Explanation: Ping Sweep scans a range of IP addresses to identify hosts that are up and responsive. It sends ICMP echo requests to discover active hosts in the specified subnet.

8. Service and Version Detection:
- Command: *`nmap -sV target_ip`*
- Explanation: Service and Version Detection is used to identify the services and versions running on open ports. Nmap sends probes to the open ports and matches responses to a database of known services and versions.

9. Port and Port Range Scanning:
- Command: `*nmap -p port(s) target_ip*`
- Explanation: Port and Port Range Scanning allows you to specify single ports or ranges to scan, helping you focus on specific services or areas of interest.

10. OS Fingerprinting:
- Command: `*nmap -O target_ip*`
- Explanation: OS Fingerprinting is used to identify the operating system of the target host. Nmap analyses responses to specific probes and matches them to known OS signatures to make an educated guess about the OS.

These Nmap scanning techniques provide various ways to identify open ports, services, and even the target's operating system, depending on the level of detail and stealth required.

OUTPUT:

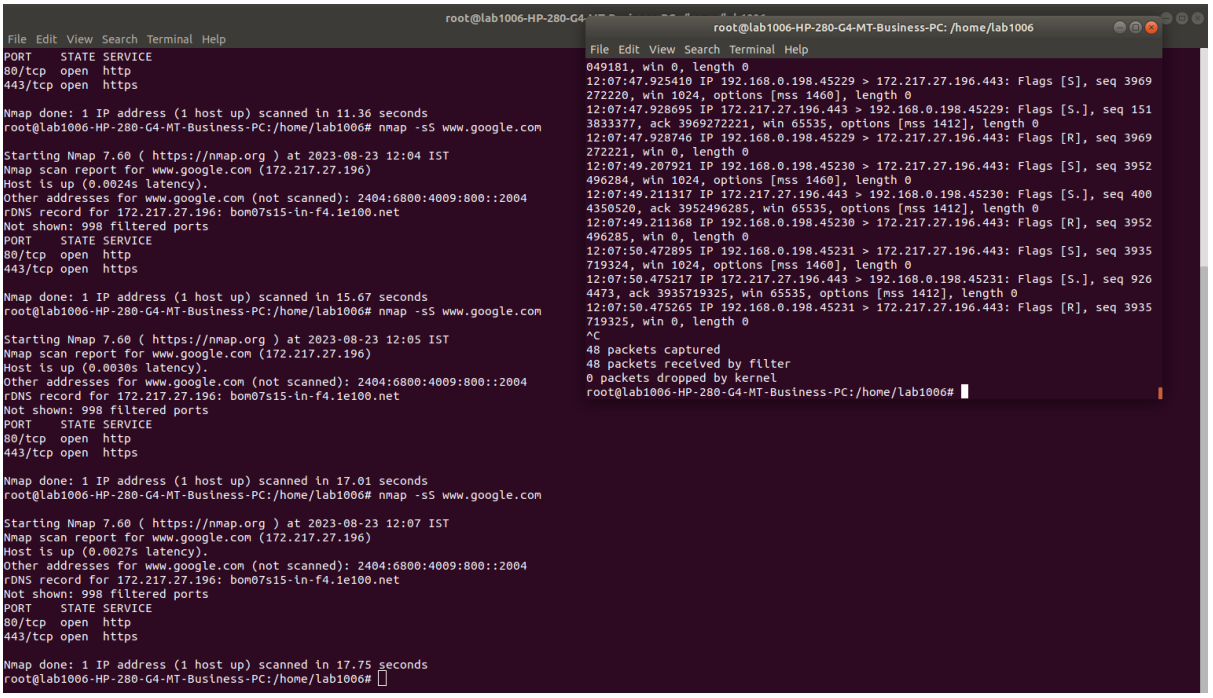
Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

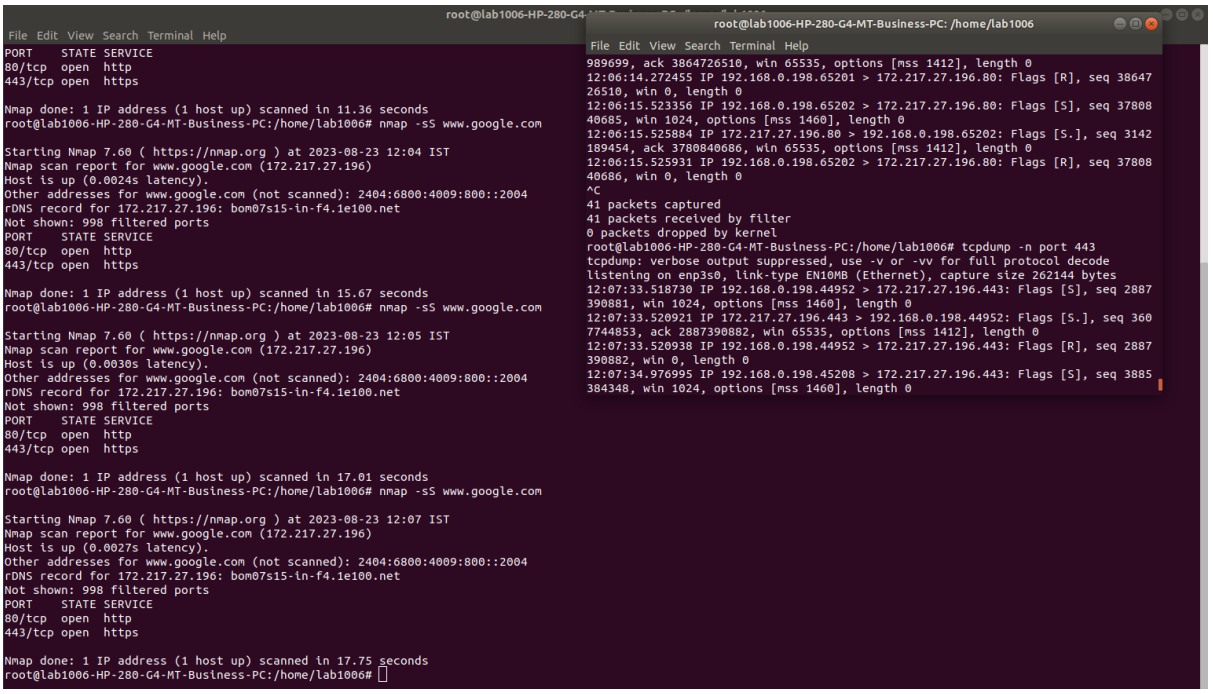
```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man nmap
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nmap -sS
You requested a scan type which requires root privileges.
QUITTING!
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo su
[sudo] password for lab1006:
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 11:59 IST
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.07 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS www.wikipedia.com

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 12:00 IST

root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS www.google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 12:02 IST
Nmap scan report for www.google.com (172.217.27.196)
Host is up (0.0025s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:800::2004
rDNS record for 172.217.27.196: bom07s15-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
File  Edit  View  Search  Terminal  Help
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS www.google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 12:04 IST
Nmap scan report for www.google.com (172.217.27.196)
Host is up (0.0024s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:800::2004
rDNS record for 172.217.27.196: bom07s15-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 15.67 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS www.google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 12:05 IST
Nmap scan report for www.google.com (172.217.27.196)
Host is up (0.0030s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:800::2004
rDNS record for 172.217.27.196: bom07s15-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 17.01 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sS www.google.com

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-23 12:07 IST
Nmap scan report for www.google.com (172.217.27.196)
Host is up (0.0027s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:800::2004
rDNS record for 172.217.27.196: bom07s15-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 17.75 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

```
root@lab1006-HP-280-G4-MT-Business-PC: /home/lab1006
File  Edit  View  Search  Terminal  Help
049181, win 0, length 0
12:07:47.925410 IP 192.168.0.198.45229 > 172.217.27.196.443: Flags [S], seq 3969
272220, win 1024, options [mss 1460], length 0
12:07:47.928695 IP 172.217.27.196.443 > 192.168.0.198.45229: Flags [S.], seq 151
3833377, ack 3969272221, win 65535, options [mss 1412], length 0
12:07:47.928746 IP 192.168.0.198.45229 > 172.217.27.196.443: Flags [R], seq 3969
272221, win 0, length 0
12:07:49.207921 IP 192.168.0.198.45230 > 172.217.27.196.443: Flags [S], seq 3952
496284, win 1024, options [mss 1460], length 0
12:07:49.211317 IP 172.217.27.196.443 > 192.168.0.198.45230: Flags [S.], seq 400
4350520, ack 3952496285, win 65535, options [mss 1412], length 0
12:07:49.211368 IP 192.168.0.198.45230 > 172.217.27.196.443: Flags [R], seq 3952
496285, win 0, length 0
12:07:50.472895 IP 192.168.0.198.45231 > 172.217.27.196.443: Flags [S], seq 3935
719324, win 1024, options [mss 1460], length 0
12:07:50.475217 IP 172.217.27.196.443 > 192.168.0.198.45231: Flags [S.], seq 926
4473, ack 3935719325, win 65535, options [mss 1412], length 0
12:07:50.475265 IP 192.168.0.198.45231 > 172.217.27.196.443: Flags [R], seq 3935
719325, win 0, length 0
^C
48 packets captured
48 packets received by filter
0 packets dropped by kernel
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.





CONCLUSION:

Hence, I have understood the fundamental concepts and carried out installation of NMAP and used it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP port scan, UDP port scan.

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

LAB ASSIGNMENT 9

Aim: Simulation of DOS attack using Hping3

Lab Outcome Attained: LO5

Theory:

Denial of Service (DoS) Attack:

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a computer network, service, or website by overwhelming it with a flood of traffic or other malicious activities. The primary objective of a DoS attack is to make the targeted system or network unavailable to its intended users, causing downtime and service disruption.

Three common types of DoS attacks are SYN flood, ICMP flood, and SMURF attack:

1. SYN Flood:

- In a SYN flood attack, the attacker sends a large number of TCP (Transmission Control Protocol) connection requests with spoofed source IP addresses to the target server.
- The target server, upon receiving these SYN (synchronise) requests, allocates resources to establish connections but never receives the expected ACK (acknowledge) response to complete the handshake.
- As a result, the server's resources, such as memory and CPU, become exhausted as it waits for acknowledgments, rendering it unable to accept legitimate connection requests and causing service disruption.

2. ICMP Flood:

- An ICMP (Internet Control Message Protocol) flood attack involves sending a massive volume of ICMP echo requests (ping requests) to a target host or network.
- The target system becomes overwhelmed by the sheer number of ICMP requests and spends resources responding to these requests.
- This flood of ICMP traffic can consume network bandwidth and processing power, causing network congestion and making the target system or network unresponsive to legitimate traffic.

3. SMURF Attack:

- A SMURF attack is a type of amplification attack that takes advantage of the ICMP protocol and IP broadcast addresses.
- The attacker sends ICMP echo requests (ping) with a spoofed source IP address to a network's broadcast address, making it appear as if the requests are originating from the target's IP address.
- All devices on the network respond to these ICMP requests, amplifying the attack and flooding the target with responses.
- This can result in a massive amount of traffic overwhelming the target's resources and causing a denial of service.

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

Hping3 Commands for SYN Flood and ICMP Flood:

Hping3 is a powerful network tool that can be used for various network testing and attack purposes. Here are example commands for performing SYN flood and ICMP flood attacks using Hping3. Please note that these commands are for educational purposes only, and using them without proper authorization is illegal and unethical.

1. SYN Flood with Hping3:

   hping3 -S -c <number_of_packets> -p <port> <target_ip>

   - -S: Indicates SYN flag for TCP packets.
   - -c <number_of_packets>: Specifies the number of packets to send.
   - -p <port>: Specifies the target port.
   - <target_ip>: The IP address of the target system.

   Example:

   hping3 -S -c 10000 -p 80 192.168.1.100

2. ICMP Flood with Hping3:

   hping3 --icmp --rand-source -c <number_of_packets> <target_ip>

   - --icmp: Specifies ICMP mode.
   - --rand-source: Randomizes the source IP address for each packet.
   - -c <number_of_packets>: Specifies the number of packets to send.
   - <target_ip>: The IP address of the target system.

   Example:

   hping3 --icmp --rand-source -c 10000 192.168.1.100

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

```
┌──(meets8㉿LAPTOP-KTF6E902)-[~]
└─$ sudo hping3 -c 4 -p 80 -i u1 192.0.2.1
[sudo] password for meets8:
HPING 192.0.2.1 (eth0 192.0.2.1): NO FLAGS are set, 40 headers + 0 data byte
s

--- 192.0.2.1 hping statistic ---
4 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(meets8㉿LAPTOP-KTF6E902)-[~]
└─$ hping3 -S -p 80 192.0.2.1
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket

┌──(meets8㉿LAPTOP-KTF6E902)-[~]
└─$ sudo hping3 -S -p 80 192.0.2.1
HPING 192.0.2.1 (eth0 192.0.2.1): S set, 40 headers + 0 data bytes
81:
^C
--- 192.0.2.1 hping statistic ---
174 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Conclusion:

It's essential to use network testing tools responsibly and only on systems or networks that you are authorised to test. Unauthorised use of such tools for malicious purposes is illegal and unethical and can result in legal consequences.

Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.

LAB ASSIGNMENT NO. 10

AIM: To study and configure Firewalls using IP tables.

LAB OUTCOME ATTAINED:

LO 6: Demonstrate the network security system using open source tools.

THEORY:

A **Firewall** is a network security device or software that monitors and controls incoming and outgoing network traffic. It acts as a barrier between a trusted internal network and untrusted external networks, such as the internet. Firewalls are designed to enforce security policies, filter traffic based on rules, and protect the network from unauthorised access, threats, and malicious activity.

Different types of firewalls include:

1. Packet Filtering Firewalls: These firewalls filter traffic based on attributes of individual network packets, such as source and destination IP addresses, port numbers, and protocol types.

2. Stateful Inspection Firewalls: Stateful firewalls keep track of the state of active connections and make decisions based on the context of the traffic. They can determine if a packet is part of an established connection and allow or deny it accordingly.

3. Proxy Firewalls: Proxy firewalls act as intermediaries between internal and external networks. They receive network requests on behalf of clients, inspect and filter the traffic, and forward it to the destination. This adds an additional layer of security.

4. Next-Generation Firewalls (NGFW): NGFWs combine traditional firewall capabilities with advanced security features such as intrusion detection, application-layer filtering, and deep packet inspection.

5. Application Layer Gateways (ALG): ALGs work at the application layer and understand specific application protocols. They can provide more granular control over application traffic.

6. Web Application Firewalls (WAF): WAFs are specialised firewalls designed to protect web applications from various web-based attacks, such as SQL injection and cross-site scripting (XSS).

7. Cloud Firewalls: Cloud providers offer firewall services for virtual machines and resources in cloud environments, allowing users to define network security rules.

Different options used in configuring a firewall can include:

- Allow Rules: Define which traffic is permitted to pass through the firewall.
- Deny Rules: Specify which traffic is blocked or rejected.
- Port-Based Rules: Control traffic based on specific ports (e.g., allowing traffic on port 80 for HTTP).
- IP Address-Based Rules: Filter traffic by source or destination IP addresses.

Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.

- Protocol Rules: Restrict traffic based on the protocol or service (e.g., allowing only FTP or SSH traffic).
- Stateful Rules: Track and allow or deny traffic based on the state of connections.
- Logging and Monitoring: Configure logging rules to keep records of allowed and denied traffic for auditing and analysis.
- Security Groups: In cloud environments, security groups are used to control inbound and outbound traffic to resources.

Commands for configuring a firewall using IPTABLES

1. To list existing rules:
   iptables -L

2. To allow incoming traffic on a specific port (e.g., port 80 for HTTP):
   iptables -A INPUT -p tcp --dport 80 -j ACCEPT

3. To deny incoming traffic on a specific port (e.g., port 22 for SSH):
   iptables -A INPUT -p tcp --dport 22 -j DROP

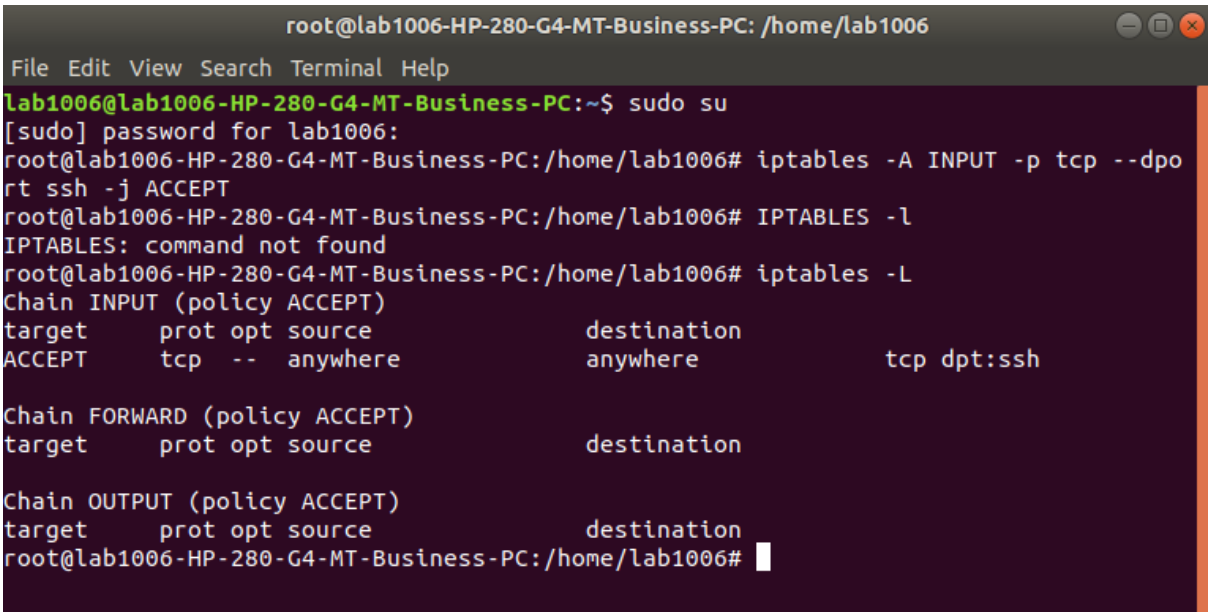
4. To save the rules:
   service iptables save

5. To restart the firewall:
  service iptables restart

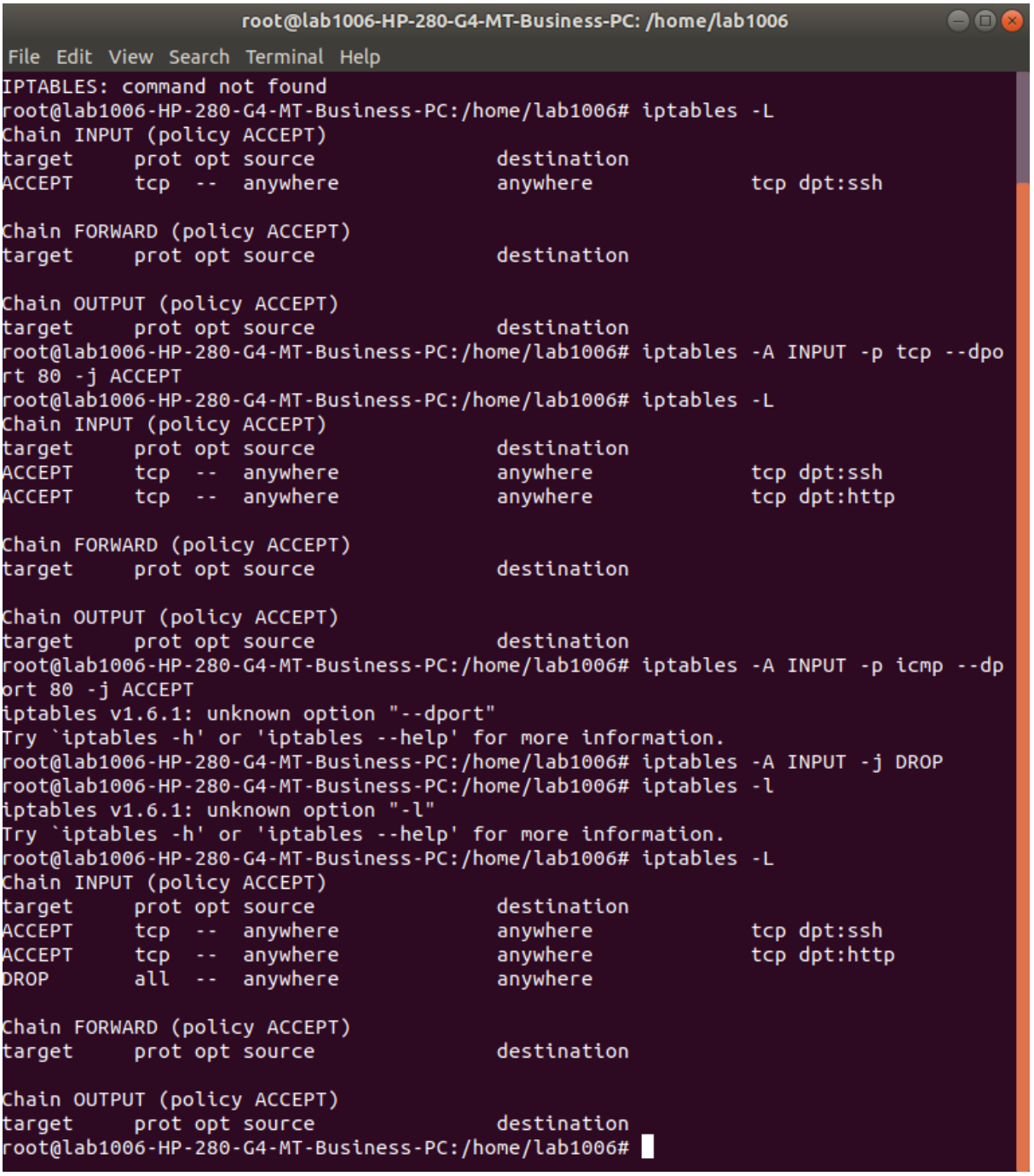
These commands are just examples, and configuring a firewall with IPTABLES can be complex and requires careful consideration of security policies and network requirements. It's essential to understand the potential impact of firewall rules on your network.

Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.

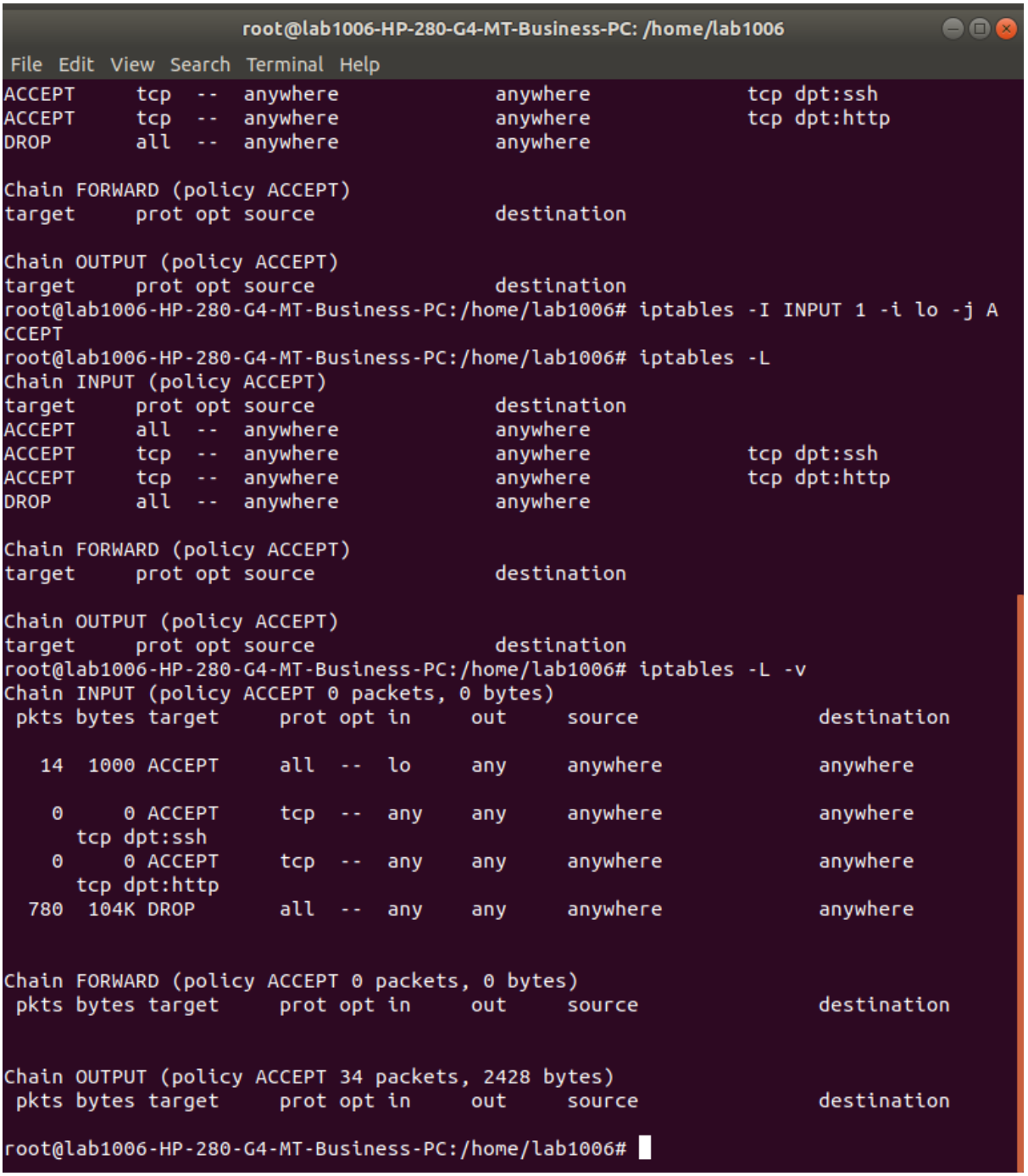
SCREENSHOTS:



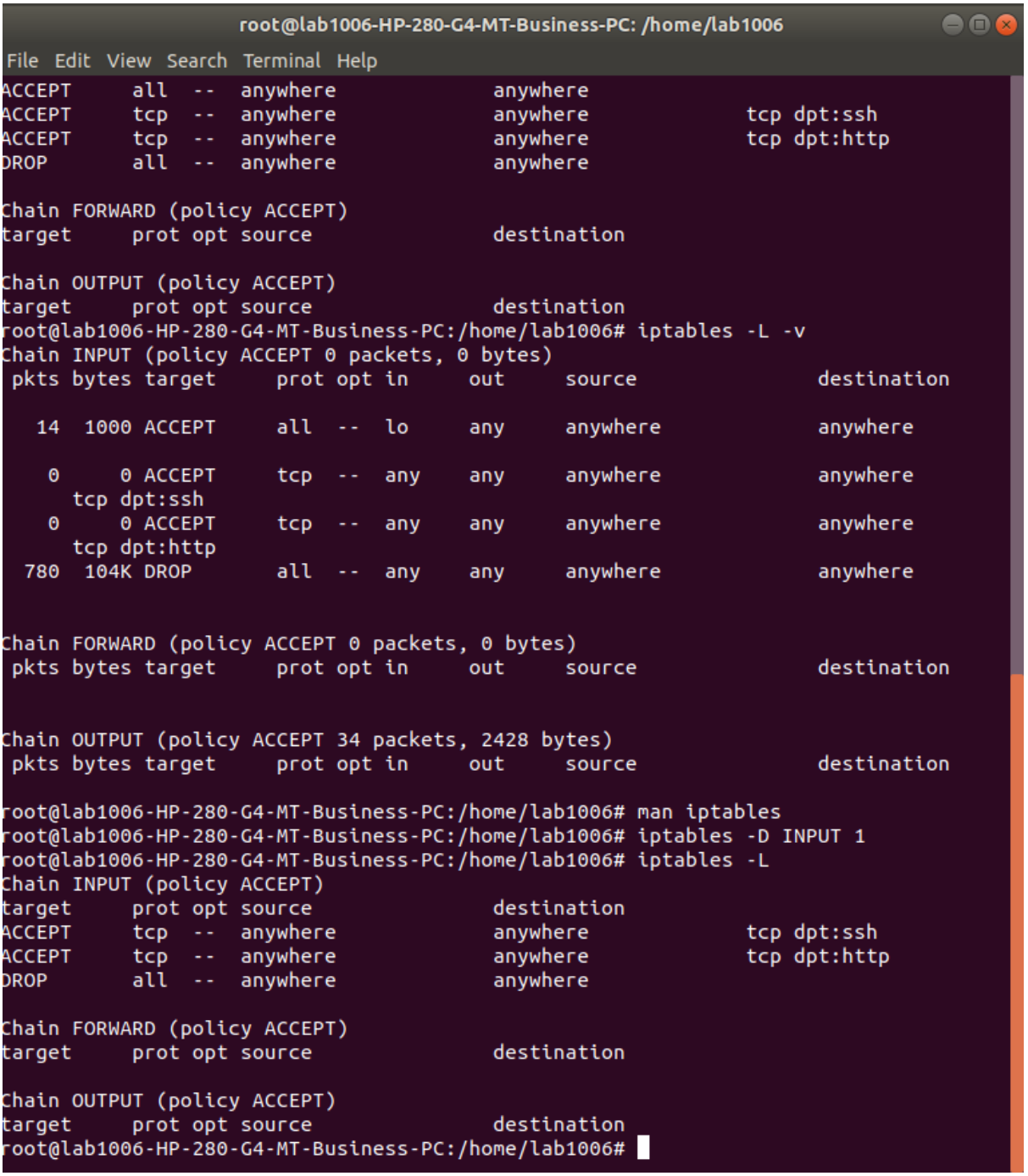
Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.



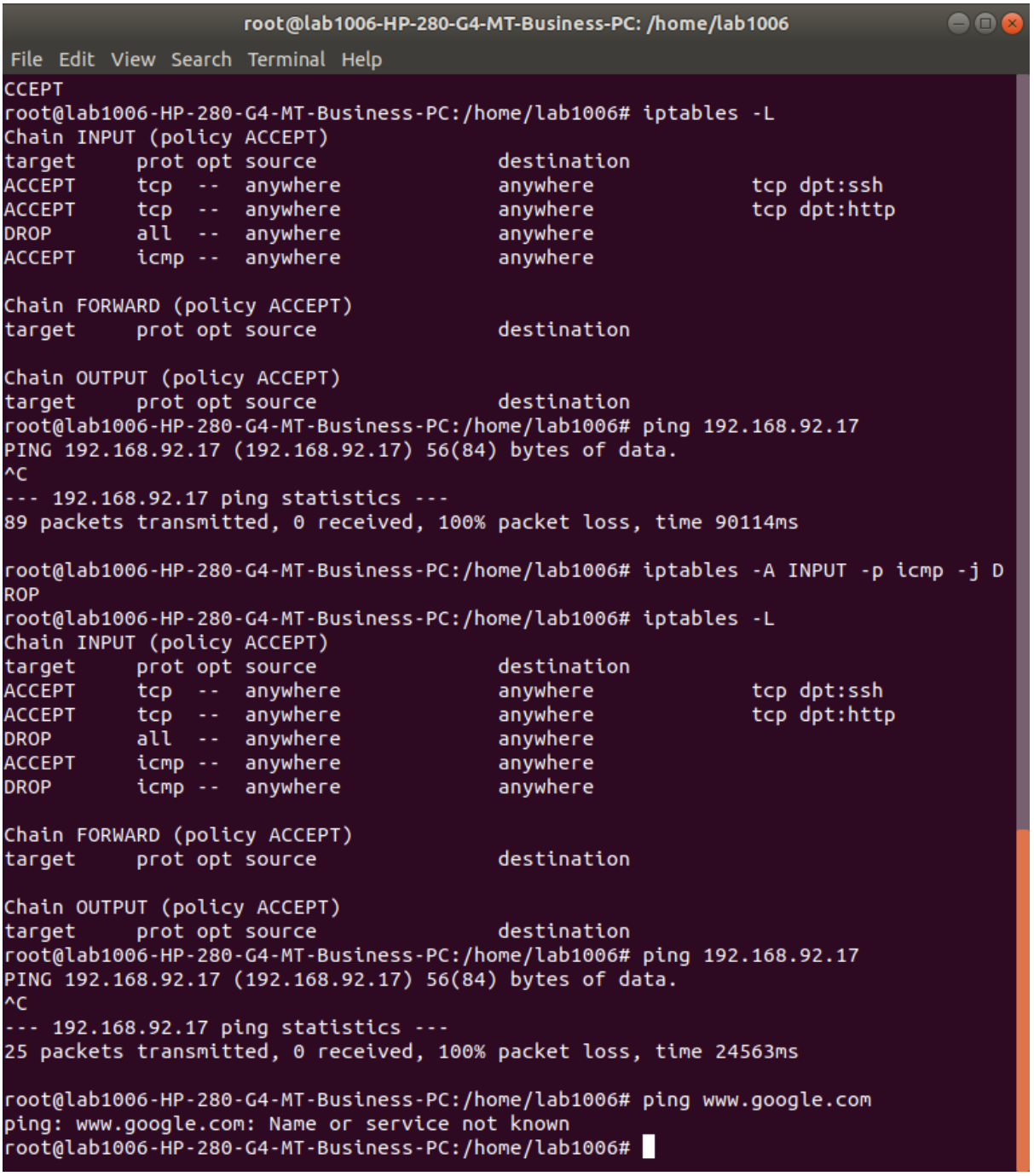
Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.

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

Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 19th October, 2023.

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere
DROP       icmp --  anywhere             anywhere
DROP       tcp  --  anywhere             anywhere
^[[A^[[A^C
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# iptables -F
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# ping 192.168.0.153
PING 192.168.0.153 (192.168.0.153) 56(84) bytes of data.
64 bytes from 192.168.0.153: icmp_seq=1 ttl=64 time=0.455 ms
64 bytes from 192.168.0.153: icmp_seq=2 ttl=64 time=0.529 ms
64 bytes from 192.168.0.153: icmp_seq=3 ttl=64 time=0.538 ms
64 bytes from 192.168.0.153: icmp_seq=4 ttl=64 time=0.527 ms
64 bytes from 192.168.0.153: icmp_seq=5 ttl=64 time=0.527 ms
64 bytes from 192.168.0.153: icmp_seq=6 ttl=64 time=0.528 ms
64 bytes from 192.168.0.153: icmp_seq=7 ttl=64 time=0.534 ms
64 bytes from 192.168.0.153: icmp_seq=8 ttl=64 time=0.576 ms
64 bytes from 192.168.0.153: icmp_seq=9 ttl=64 time=0.528 ms
64 bytes from 192.168.0.153: icmp_seq=10 ttl=64 time=0.532 ms
64 bytes from 192.168.0.153: icmp_seq=11 ttl=64 time=0.527 ms
64 bytes from 192.168.0.153: icmp_seq=12 ttl=64 time=0.572 ms
64 bytes from 192.168.0.153: icmp_seq=13 ttl=64 time=0.531 ms
64 bytes from 192.168.0.153: icmp_seq=14 ttl=64 time=0.529 ms
64 bytes from 192.168.0.153: icmp_seq=15 ttl=64 time=0.484 ms
64 bytes from 192.168.0.153: icmp_seq=16 ttl=64 time=0.573 ms
^C
--- 192.168.0.153 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15337ms
rtt min/avg/max/mdev = 0.455/0.530/0.576/0.038 ms
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

CONCLUSION:

In summary, firewalls are essential for network security, offering protection against external threats. Different firewall types and configuration options provide flexibility in safeguarding networks. Understanding firewall management tools like IPTABLES is crucial for effective rule creation and maintenance. Well-configured firewalls are fundamental in maintaining network security and enforcing access control policies.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

LAB ASSIGNMENT NO. 11

AIM: Installing snort, setting in Intrusion Detection Mode and writing rules for Intrusion Detection.

LAB OUTCOME ATTAINED:

LO 6: Demonstrate the network security system using open source tools.

THEORY:

An **Intrusion Detection System (IDS)** is a security tool or application that monitors network traffic or system activities for signs of suspicious or malicious behaviour. The primary purpose of an IDS is to detect and alert administrators or security personnel to potential security threats, policy violations, or unauthorised access attempts. IDSs work by analysing patterns in network traffic, system logs, or application behaviour, and comparing these patterns to known attack signatures or predefined rules. When suspicious activity is detected, the IDS generates alerts or takes other actions to mitigate the threat.

There are two main categories of Intrusion Detection Systems:

Network-Based IDS (NIDS): NIDS monitors network traffic and analyses data packets passing through a network segment or a specific network interface. It can detect attacks targeting network services, such as port scans, malware communication, and network-based exploits.

Host-Based IDS (HIDS): HIDS runs on individual hosts or servers and monitors activities at the operating system and application levels. It can detect attacks targeting the host, such as unauthorised file access, suspicious system calls, or changes to critical system files.

Snort is a widely used open-source Intrusion Detection System (IDS) that can operate in various modes to suit different security needs. Here are the **different modes** in which Snort can work:

1. **Sniffer Mode**: In this mode, Snort behaves like a packet sniffer, capturing network traffic and displaying it on the console. This mode is primarily used for testing and troubleshooting purposes.
2. **Packet Logger Mode**: In this mode, Snort logs captured packets to a binary file without performing any analysis. Later, these packets can be analysed using other tools. This mode is useful for packet capture and forensic analysis.
3. **Network Intrusion Detection Mode (NIDS)**: This is the primary mode for Snort as an IDS. In NIDS mode, Snort monitors network traffic, analyses packets, and compares them against predefined rules or signatures to detect and alert on suspicious or malicious activity.
4. **Inline Mode (IPS)**: Snort can be configured to operate in Inline Intrusion Prevention System (IPS) mode. In this mode, Snort not only detects intrusions but also can take action to block or drop suspicious traffic, effectively acting as a firewall.

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

Commands for installing Snort, editing its configuration file, and configuring it in Intrusion Detection Mode (NIDS):

Installation:

sudo apt-get install snort

Editing the Configuration File (modify as needed):

The Snort configuration file is typically located at /etc/snort/snort.conf. You can edit it with a text editor:

sudo nano /etc/snort/snort.conf

Customise the configuration file according to your network setup, rules, and requirements.

Starting Snort in NIDS Mode:

To start Snort in NIDS mode, you can use the following command:

sudo snort -A alert -q -u snort -g snort -c /etc/snort/snort.conf -i <your_network_interface>

-A alert: Specifies the alert mode for generating alerts.

-q: Runs Snort in quiet mode to reduce console output.

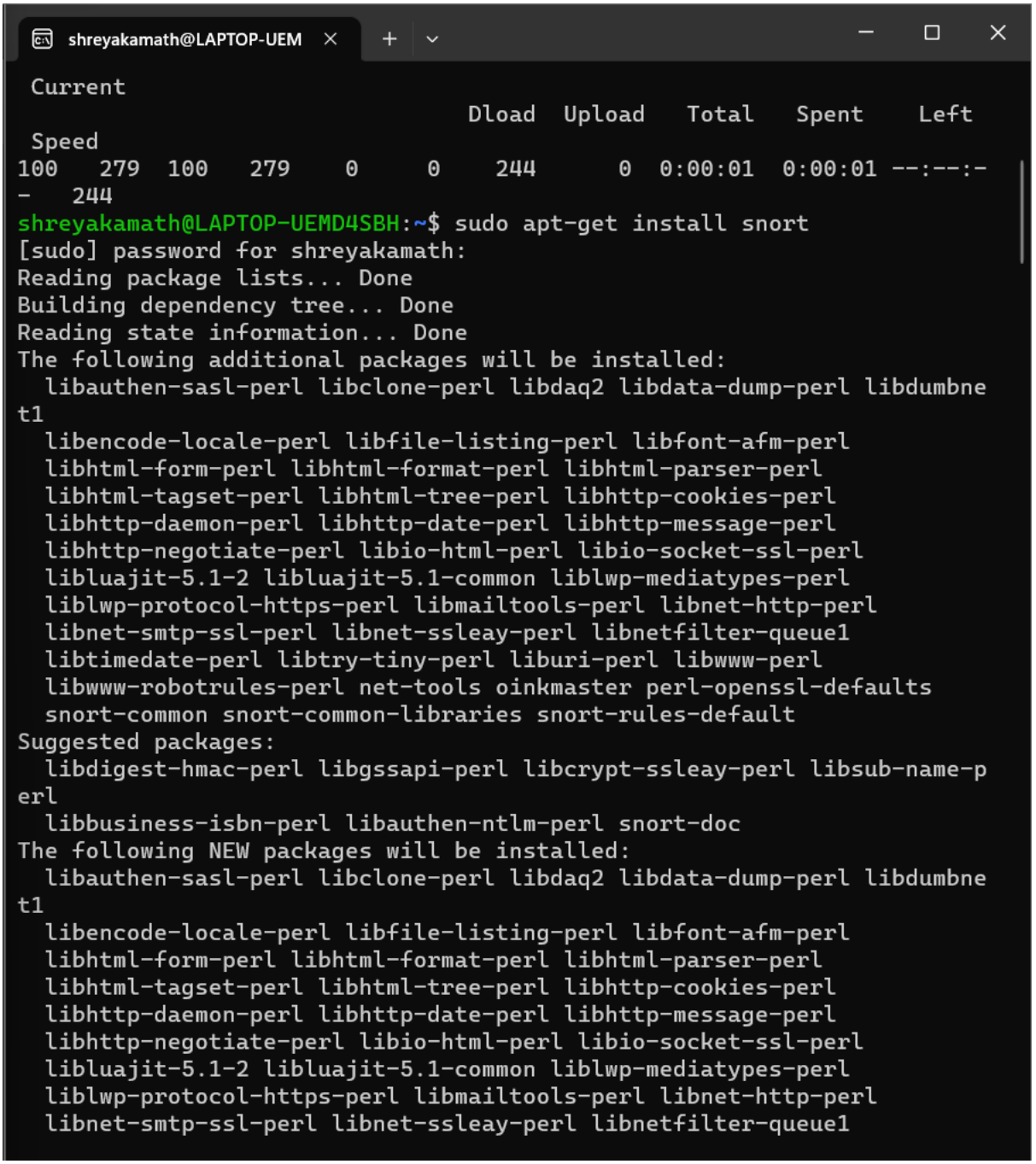
-u and -g: Specify the user and group to run Snort under.

-c: Points to the Snort configuration file.

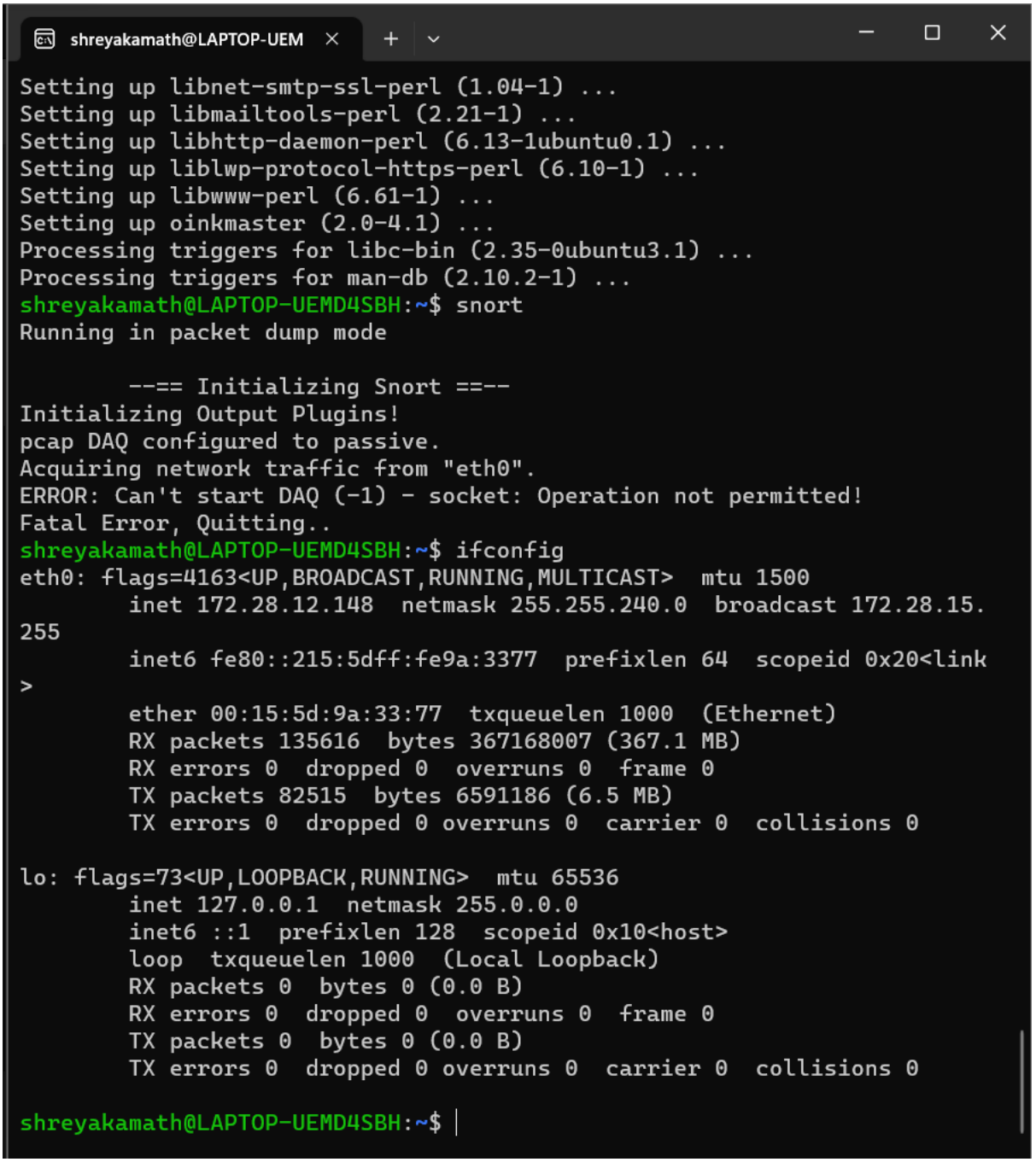
-i: Specifies the network interface to monitor.

OUTPUT:

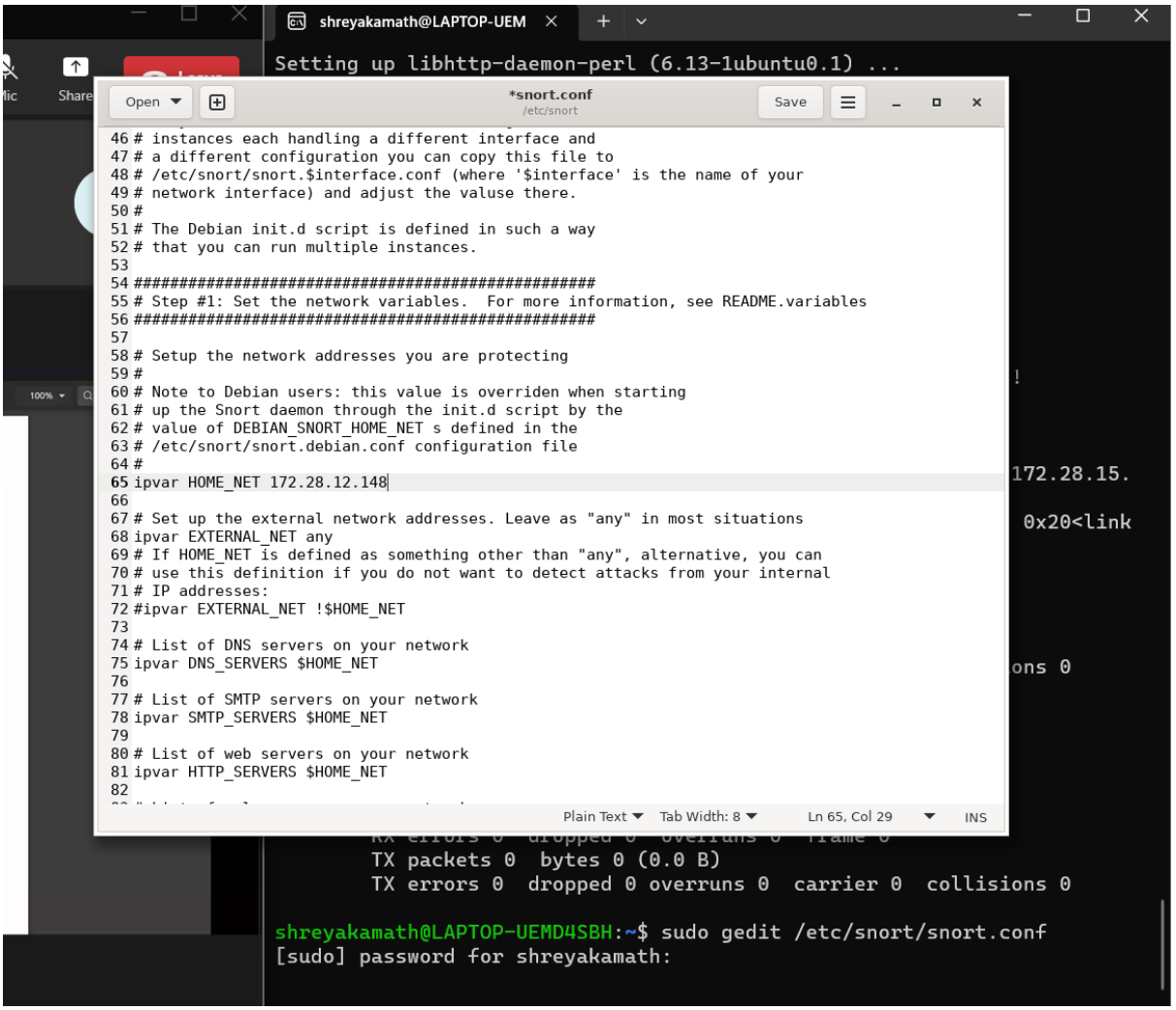
Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.



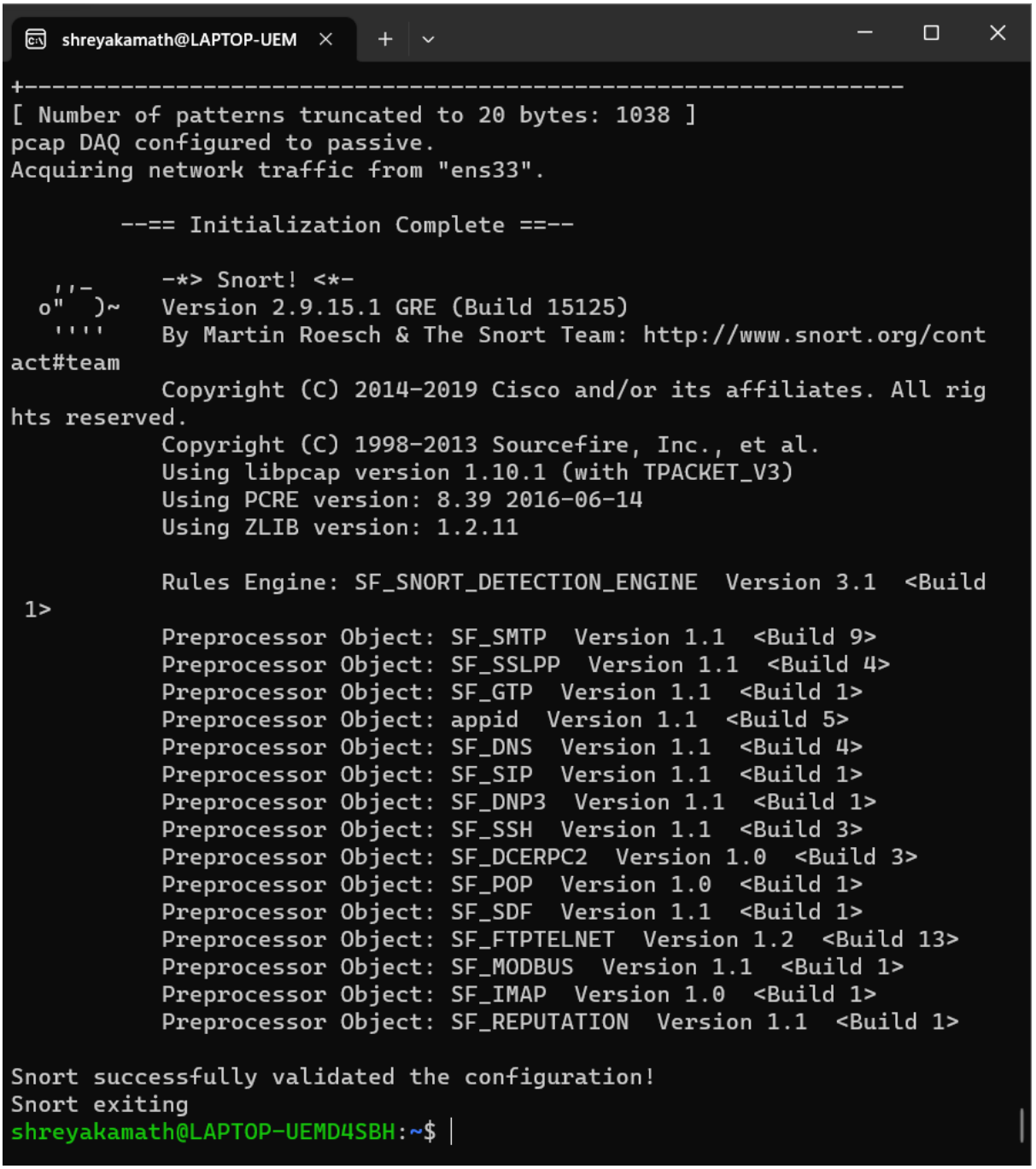
Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.

```
shreyakamath@LAPTOP-UEMD4SBH:~$ sudo snort -T -c /etc/snort/snort.conf
-i ens33
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1
741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 71
44:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 812
3 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443
 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 90
1 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988
 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088
8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9
090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
```

Roll no. : 53
Name: Shreya Kamath
Date: 12th September, 2023.



CONCLUSION:

Hence, I have learnt the fundamentals of Intrusion Detection Systems and Snort and its various modes and commands and executed them on a WSL2 Ubuntu Virtual Machine.

Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.

LAB ASSIGNMENT NO. 12

AIM: Explore the GPG tool of linux to implement email security.

LAB OUTCOME ATTAINED:

LO 6: Demonstrate the network security system using open source tools.

THEORY:

GPG, which stands for **GNU Privacy Guard,** is a free and open-source encryption software that provides cryptographic privacy and authentication for data communication. It's a modern replacement for PGP (Pretty Good Privacy) and is widely used for securing emails and files. GPG is available for various platforms, including Linux, macOS, and Windows. In GPG, the terms "private key ring" and "public key ring" refer to collections of cryptographic keys used for encryption and decryption:

Private Key Ring:

- The private key ring is a file or storage location where the sender's private keys are stored.
- Private keys are used by the sender for decrypting messages that were encrypted with the receiver's public key and for digitally signing messages or files.
- The sender should guard their private key(s) very carefully because anyone with access to the private key can decrypt their messages and files, impersonate them, and potentially compromise their security.

Public Key Ring:

- The public key ring is a file or storage location where the public keys of the receiver and other potential recipients are stored.
- Public keys are used by the sender for encrypting messages or files that they want to send securely to the receiver. The sender uses the receiver's public key to encrypt the data, and only the receiver can decrypt it with their private key.
- Additionally, public keys are used to verify digital signatures created by the sender or others. If the sender receives a digitally signed message or file, they can use the sender's public key to verify that it was indeed signed by them and hasn't been tampered with.

Commands used for Key Generation and Encryption/Decryption:

Step 1: Generate private key and public key pairs for sender and receiver using command
gpg --gen-key or gpg –full-generate-key (repeat for sender and receiver)

Step 2: Create a file containing sender’s public key which then can be sent to other users.
gpg --export -a username>filename (creates file in ascii format) or
gpg --output filename --armor --export user’s_email (for sender)

Step 3: Similarly create a file containing the sender's private key.
gpg --export-secret-key -a username>filename (for sender)

Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.

Step 4: You can create a fingerprint of key using the command
gpg --fingerprint receiver's_email (for receiver)

Step 5: Sender needs to add in his public key ring, the public key of receiver (for sender)
gpg --import filename_containing_public_key_of_receiver

Step 6: Listing public keys in keyring
gpg --list-keys (from public key rings of all users)
gpg --list-keys emailid@gmail.com (from public key rings of specific users)

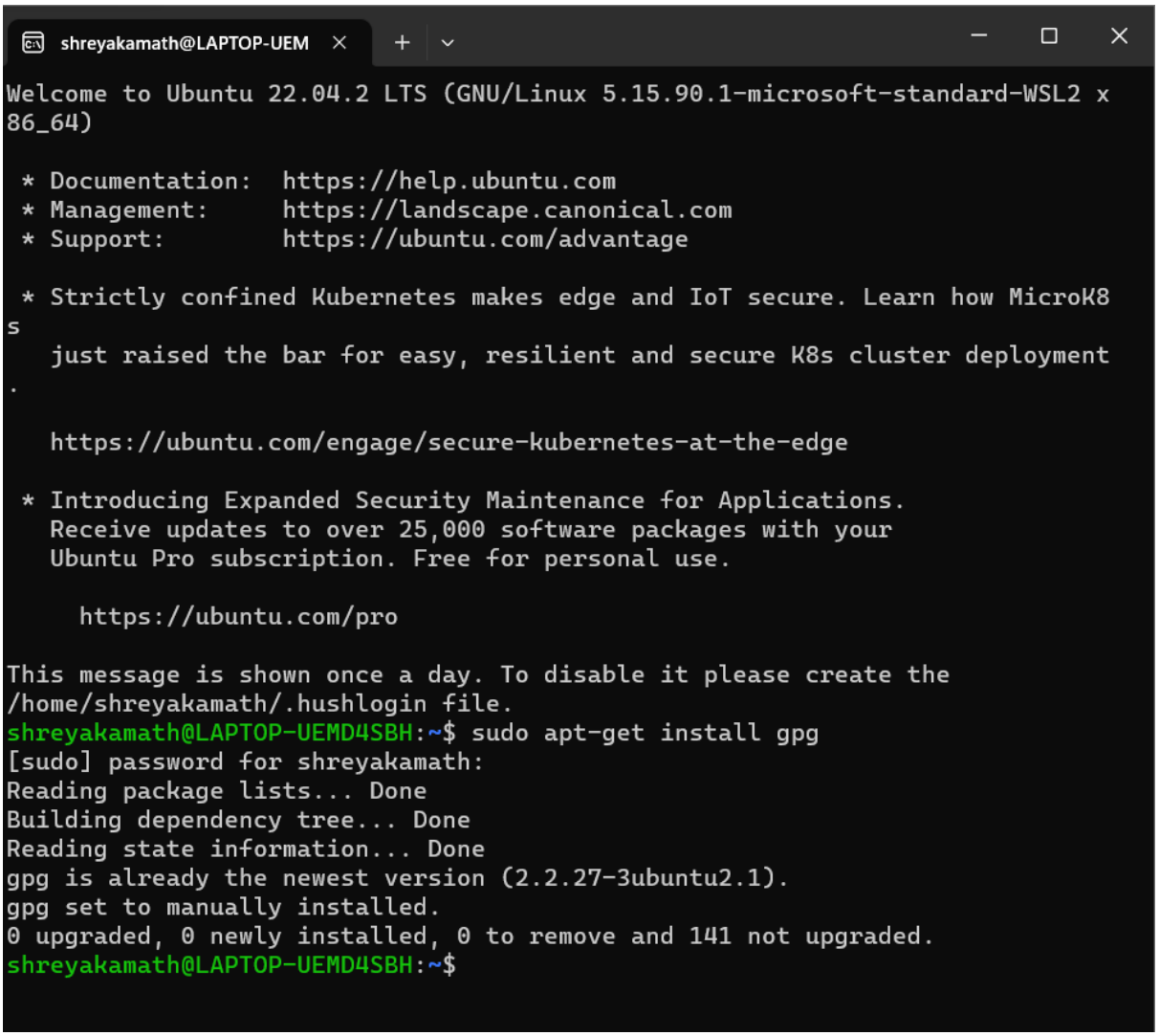
Step 7: Sender can sign the public key of receiver using command
gpg --sign-key receiver_email

Step 8: Encrypt the data to send.
gpg --encrypt -r receiver_email name_of_file
OR
gpg --encrypt --sign --armor -r receiver_email name_of_file
OR
gpg --encrypt --sign -r receiver_email name_of_file

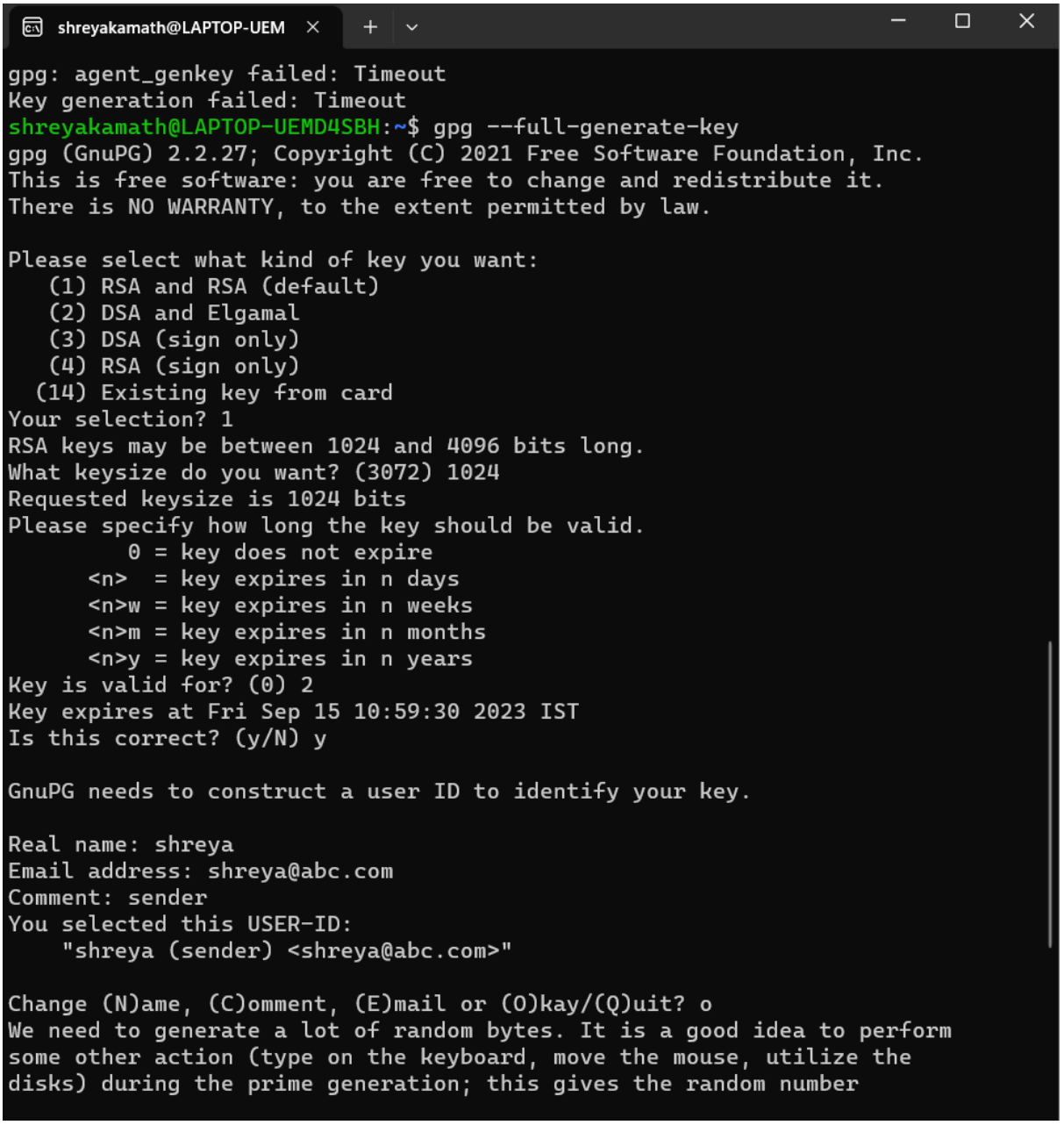
Step 9: Decrypt the file
gpg -o myfiledecrypted -d myfile.txt.gpg

Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.

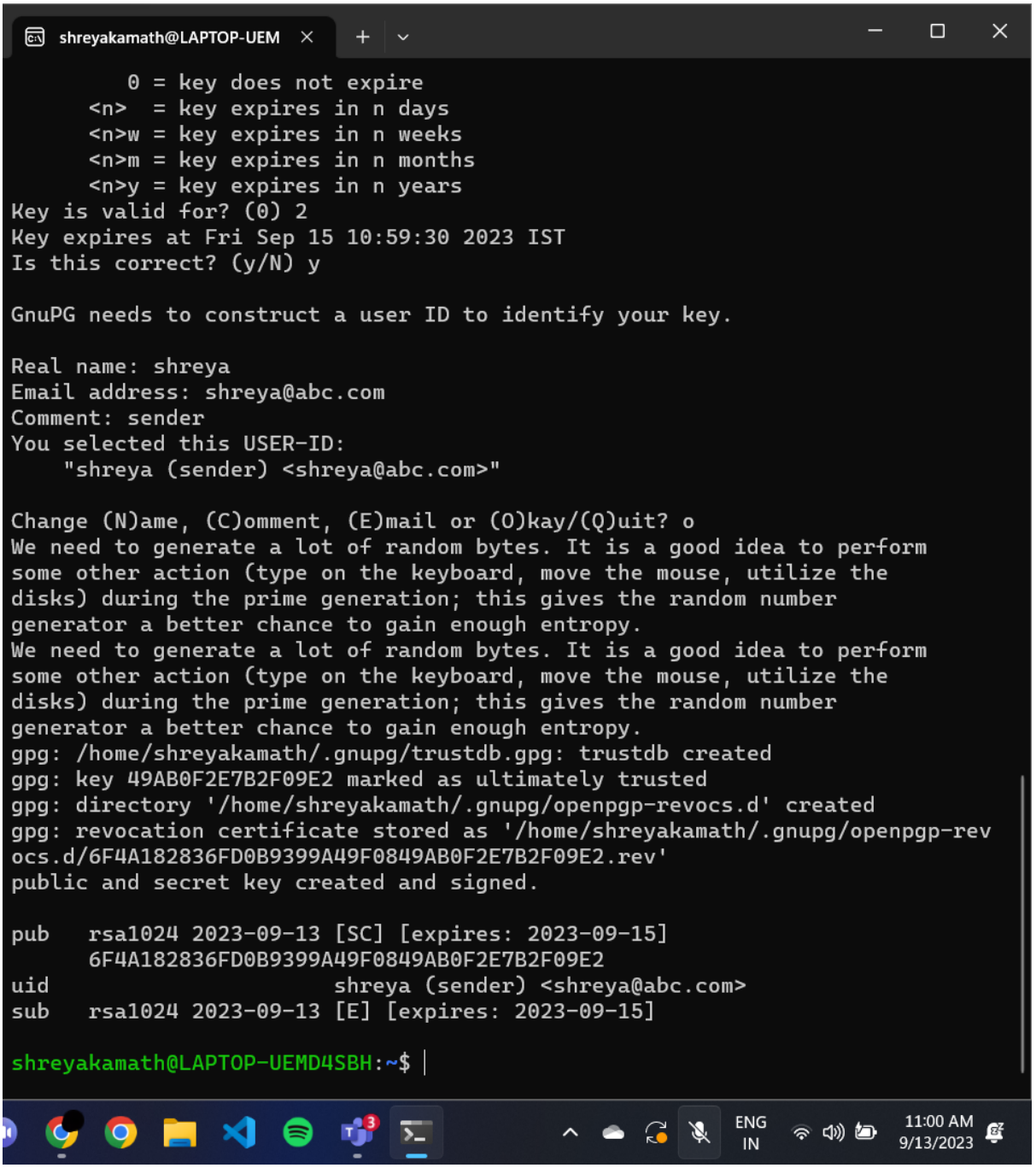
OUTPUT:



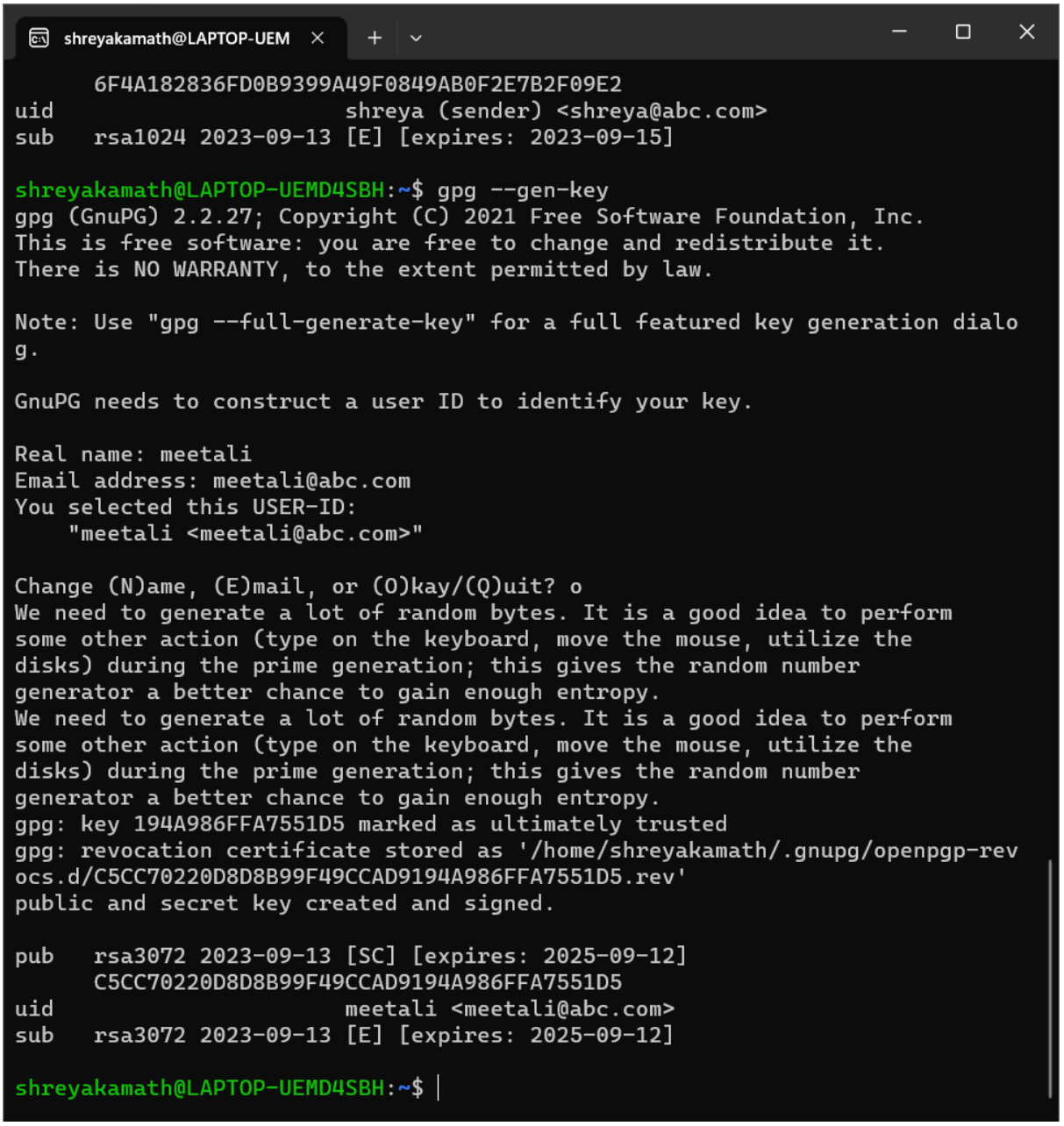
Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.



Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.



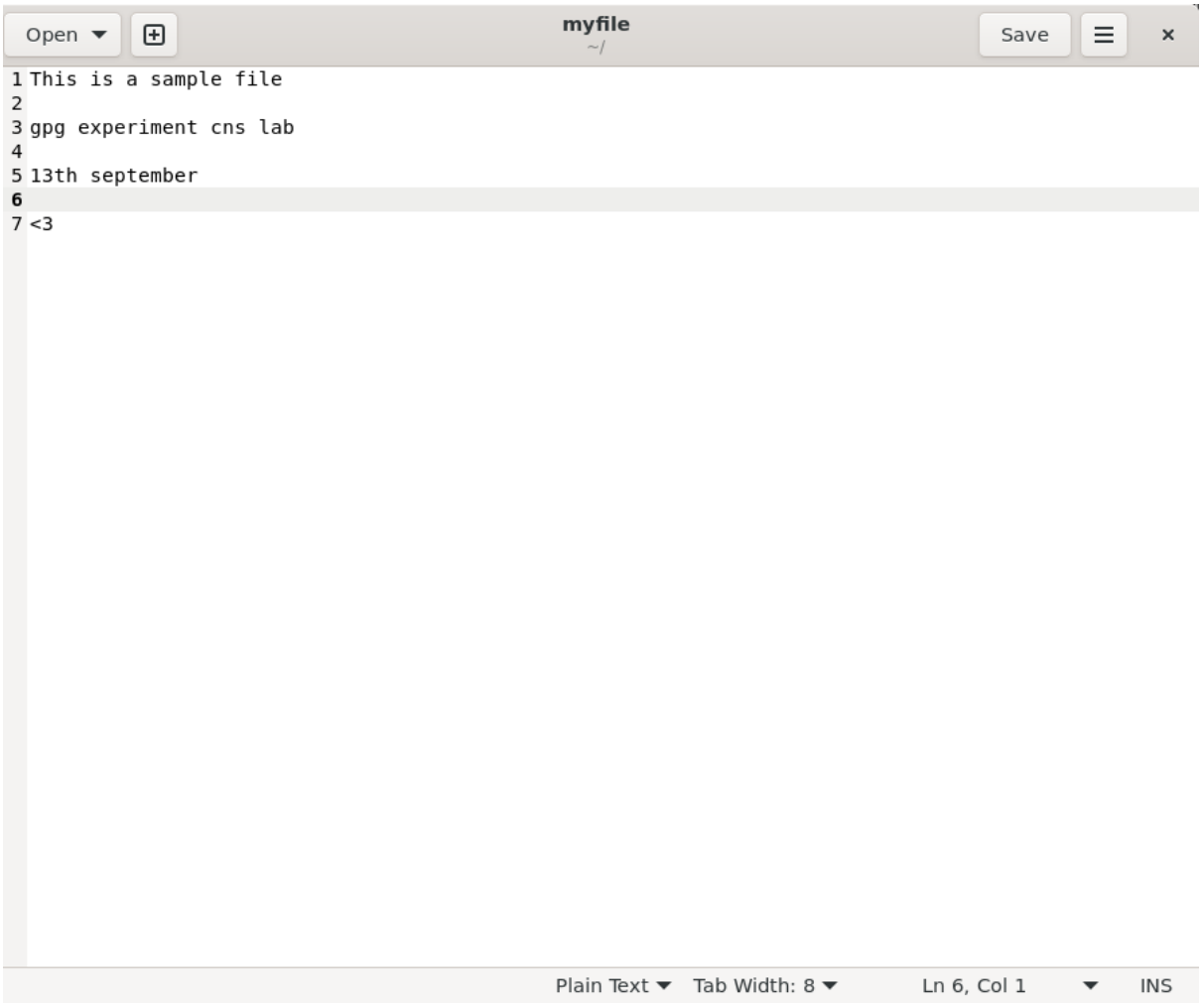
Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.

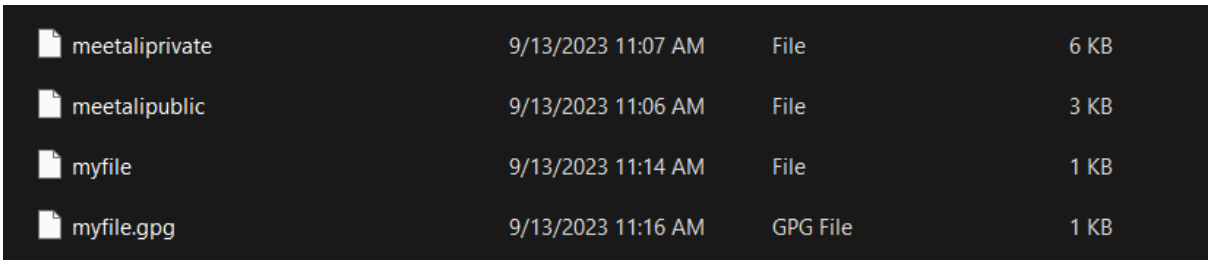
```
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --export -a shreya>shreyapublic
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --output meetalipublic --armor --export
meetali@abc.com
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --export-secret-key -a shreya>shreyapriv
ate
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --export-secret-key -a meetali>meetalipr
ivate
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --import meetalipublic
gpg: key 194A986FFA7551D5: "meetali <meetali@abc.com>" not changed
gpg: Total number processed: 1
gpg:               unchanged: 1
shreyakamath@LAPTOP-UEMD4SBH:~$
```

```
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --import meetalipublic
gpg: key 194A986FFA7551D5: "meetali <meetali@abc.com>" not changed
gpg: Total number processed: 1
gpg:               unchanged: 1
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2023-09-15
/home/shreyakamath/.gnupg/pubring.kbx
-------------------------------------
pub   rsa1024 2023-09-13 [SC] [expires: 2023-09-15]
      6F4A182836FD0B9399A49F0849AB0F2E7B2F09E2
uid           [ultimate] shreya (sender) <shreya@abc.com>
sub   rsa1024 2023-09-13 [E] [expires: 2023-09-15]

pub   rsa3072 2023-09-13 [SC] [expires: 2025-09-12]
      C5CC70220D8D8B99F49CCAD9194A986FFA7551D5
uid           [ultimate] meetali <meetali@abc.com>
sub   rsa3072 2023-09-13 [E] [expires: 2025-09-12]

shreyakamath@LAPTOP-UEMD4SBH:~$
```

```
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --list-keys shreya@abc.com
pub   rsa1024 2023-09-13 [SC] [expires: 2023-09-15]
      6F4A182836FD0B9399A49F0849AB0F2E7B2F09E2
uid           [ultimate] shreya (sender) <shreya@abc.com>
sub   rsa1024 2023-09-13 [E] [expires: 2023-09-15]

shreyakamath@LAPTOP-UEMD4SBH:~$ gedit myfile
```

Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.





Roll no. : 53
Name: Shreya Kamath
Date: 13th September, 2023.

```
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt -r meetali@abc.com myfile
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt --sign --armor -r meetali@abc.
com
^C
gpg: signal 2 caught ... exiting

shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt --sign --armor -r meetali@abc.
com myfile
shreyakamath@LAPTOP-UEMD4SBH:~$
```

```
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt -r meetali@abc.com myfile
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt --sign --armor -r meetali@abc.
com
^C
gpg: signal 2 caught ... exiting

shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt --sign --armor -r meetali@abc.
com myfile
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg --encrypt --sign -r meetali@abc.com myfi
le
File 'myfile.gpg' exists. Overwrite? (y/N) y
shreyakamath@LAPTOP-UEMD4SBH:~$ gpg -o myfiledecrypted -d myfile.gpg
gpg: encrypted with 3072-bit RSA key, ID 7683BC5E819F5859, created 2023-09-1
3
      "meetali <meetali@abc.com>"
gpg: Signature made Wed Sep 13 11:21:20 2023 IST
gpg:                using RSA key 6F4A182836FD0B9399A49F0849AB0F2E7B2F09E2
gpg: Good signature from "shreya (sender) <shreya@abc.com>" [ultimate]
shreyakamath@LAPTOP-UEMD4SBH:~$
```

meetaliprivate	9/13/2023 11:07 AM	File	6 KB
meetalipublic	9/13/2023 11:06 AM	File	3 KB
myfile	9/13/2023 11:14 AM	File	1 KB
myfile.asc	9/13/2023 11:18 AM	ASC File	2 KB
myfile.gpg	9/13/2023 11:21 AM	GPG File	1 KB
myfiledecrypted	9/13/2023 11:23 AM	File	1 KB

CONCLUSION:

Hence, I have understood the basic concept and fundamentals of the GPG tool for email security purposes. I also executed related commands for encryption, decryption and key generation using the GPG tool.

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

CNS Theory Assignment 1

Q. Explain the padding scheme used in RSA. Why is it used? What is its limitation?

Padding in RSA (Rivest-Shamir-Adleman) is a critical component of the encryption and decryption processes. It is used to address vulnerabilities and limitations inherent in the basic RSA algorithm. The primary goals of padding in RSA are to ensure security, correctness, and uniqueness in encrypted messages. The two most commonly used padding schemes in RSA are PKCS#1 v1.5 and OAEP (Optimal Asymmetric Encryption Padding).

The Purpose and Importance of Padding in RSA:

- Enhanced Security: Padding bolsters the security of RSA by making it more resilient to various attacks, including chosen plaintext attacks and ciphertext attacks. Without proper padding, RSA can be vulnerable to attacks that exploit mathematical properties of the RSA algorithm, such as its homomorphic property.

- Correctness Assurance: Padding ensures that the plaintext message is correctly retrieved after decryption. Without padding, the recipient might struggle to distinguish the actual message from random noise.

- Uniqueness Guarantee: Padding ensures that distinct plaintexts produce distinct ciphertexts, even if they have identical numeric values when converted into integers. This is vital to prevent attackers from detecting patterns in the ciphertext.

PKCS#1 v1.5 Padding Scheme:

PKCS#1 v1.5 padding is among the most widely used padding schemes for RSA and involves the following steps:

1. Padding Generation:
   - Calculate the padding length 'k' as follows: k = n/8 - 3, where 'n' is the RSA modulus size in bits.
   - Generate 'k' random bytes to create the padding (P).
   - Concatenate the following components to form the padded message (PM): 0x00 || 0x02 || P || 0x00 || M. Here, 0x00 is a byte with all bits set to zero, and 0x02 serves as a marker byte.

2. Encryption:
   - Convert the padded message (PM) into an integer (m).
   - Encrypt 'm' using the RSA public key (e, n): c = m^e mod n.

PKCS#1 v1.5 padding is relatively straightforward and widely supported but has some limitations. It is vulnerable to a padding oracle attack, and the padding overhead can be significant for short messages, affecting the efficiency of RSA encryption.

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

Optimal Asymmetric Encryption Padding (OAEP):

OAEP is a more advanced padding scheme designed to overcome the limitations of PKCS#1 v1.5 padding. It offers improved security against chosen plaintext attacks and padding oracle attacks. OAEP includes these steps:

1. Padding Generation:
   - OAEP uses two hash functions, typically H (e.g., SHA-256) and a mask generation function G.
   - Calculate the maximum message length that can be encrypted with OAEP.
   - For a plaintext message (M) of length (mLen), perform the following steps:
     - Generate a random k0-bit string called 'seed.'
     - Calculate 'dbMask' by applying G to 'seed.'
     - XOR 'M' with 'dbMask' to create the data block 'DB.'
     - Compute 'seedMask' by applying G to 'DB.'
     - XOR 'seed' with 'seedMask' to create the masked seed 'maskedSeed.'
     - Concatenate 'maskedSeed' and 'DB' to create the padded message 'EM.'

2. Encryption:
   - Convert the padded message 'EM' into an integer (m).
   - Encrypt 'm' using the RSA public key (e, n): c = m^e mod n.

In summary, padding in RSA is vital for improving security, correctness, and the uniqueness of encrypted messages. While PKCS#1 v1.5 is widely used, OAEP is a more secure alternative that mitigates vulnerabilities, making it a preferred choice for modern RSA encryption applications.

Roll no. : 53
Name: Shreya Kamath
Date: 22nd September, 2023.

Theory Assignment 2

Q. What is an Intrusion Detection System? Explain different types of intrusion detection systems with their working. State the advantages and limitations of each.

An Intrusion Detection System (IDS) serves as a valuable security tool, designed to monitor network traffic or system activities to uncover any unauthorized or malicious actions within a network or system. It adds an extra layer of security, helping to identify potential security breaches, policy violations, or irregularities.

The primary types of Intrusion Detection Systems include:

1. Network-based Intrusion Detection System (NIDS):

- NIDS continuously observes network traffic in real-time and analyzes data packets to identify suspicious or malicious activities.
- It employs predefined signatures, rules, or algorithms to compare network traffic against known attack patterns.
- Upon identifying a match, NIDS generates alerts or enacts responses such as logging or blocking the event.

Advantages:

- Provides Network-Wide Visibility: NIDS monitors the entire network, making it effective for detecting attacks targeting multiple systems.
- Real-time Monitoring: It operates in real-time, allowing for swift detection and response to ongoing attacks.
- Centralized Management: NIDS offers centralized control for monitoring and managing network traffic, simplifying the administration of security policies.

Limitations:

- Struggles with Encrypted Traffic: Analyzing encrypted traffic can pose a challenge, reducing its effectiveness against attacks within encrypted communication channels.
- Overwhelmed by High Traffic Volume: High volumes of traffic can overwhelm NIDS, leading to missed detections or false positives.
- Potential for False Positives: NIDS may produce false alerts, necessitating additional investigation by security personnel.

2. Host-based Intrusion Detection System (HIDS):

- HIDS is installed on individual hosts (such as computers or servers) and focuses on monitoring activities specific to each host, including file system changes, log analysis, and system calls.
- It compares observed activities with predefined patterns or rules to detect anomalies or potential intrusions.
- Alerts are generated if observed behavior deviates from established rules.

Advantages:

- Offers Granular Visibility: HIDS provides detailed insights into activities on a specific host, enabling in-depth analysis and monitoring.
- Effective Against Insider Threats: It excels at detecting insider threats or unauthorized activities that may not traverse the network.
- Supports Compliance Monitoring: HIDS aids in compliance with security policies and regulations by monitoring host-level activities.

Limitations:

- Limited in Scope: HIDS monitors activities only on the host where it is installed, making it less effective for detecting attacks involving multiple hosts.
- Resource Overhead: Depending on the level of monitoring, HIDS can consume significant system resources (CPU and memory), potentially affecting host performance.
- Dependency on Local Logs: It relies on local log files and system events, which can be manipulated or deleted by attackers.

3. Anomaly-based Intrusion Detection System:

- Anomaly-based IDS focuses on establishing a baseline of normal behavior for a system or network.
- It continuously monitors and assesses system or network activities, searching for deviations from the established baseline.
- When an anomaly or unusual behavior is detected, the system triggers an alert or takes appropriate action.

Advantages:

- Detects Unknown Threats: Anomaly-based IDS can identify previously unknown or zero-day attacks that lack known signatures.
- Adaptive: It can adapt to changing network or system conditions and recalibrate the baseline accordingly.
- Low False Positives: Anomaly-based IDS tends to generate fewer false positives as it doesn't rely on predefined signatures.

Limitations:

- Complex Baseline Establishment: Setting up an accurate baseline can be challenging, and the system may produce false alarms during the initial learning phase.
- Difficulty in Defining Normal Behavior: Defining "normal" behavior can be subjective and may vary depending on the environment.

4. Hybrid Intrusion Detection System:

- A Hybrid IDS amalgamates components of multiple IDS types (e.g., NIDS, HIDS, and Anomaly-based IDS) to enhance detection accuracy and coverage.
- It leverages the strengths of each IDS type to provide a more comprehensive and effective intrusion detection solution.
- For instance, it may use NIDS for network-level monitoring, HIDS for host-level monitoring, and anomaly-based detection to spot unusual patterns.

Advantages:

- Enhanced Detection: Combining different IDS types can lead to improved detection capabilities by compensating for each type's limitations.
- Customization: Organisations can tailor the hybrid IDS to meet their specific security requirements and infrastructure.

Limitations:

- Complexity: Setting up and managing a hybrid IDS can be complex and may require more resources and expertise.
- Cost: The cost of implementing and maintaining multiple IDS components can be higher than deploying a single IDS type.

These four primary Intrusion Detection System types, namely NIDS, HIDS, Anomaly-based IDS, and Hybrid IDS, provide various approaches to detecting and mitigating security threats. The selection of which type(s) to employ hinges on an organization's security needs, available resources, and the specific threats they aim to safeguard against. Often, organizations opt for a combination of these IDS types to establish a multi-layered defense against intrusion attempts and cyberattacks.

CNS PAPER PRESENTATION

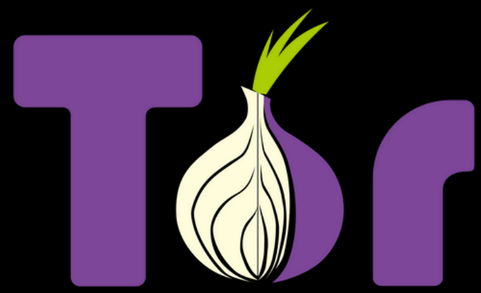
A Review of Dark Web: Trends and Future Directions

Shahriar Sobhan, Timothy Williams, Md Jobair Hossain Faruk, Juan Rodriguez
Masrura Tasnim, Edwin Mathew, Jack Wright & Hossain Shahriar

VAISHNAVI HINDALEKAR (40)
SHREYA KAMATH (53)
MEETALI KAPSE (54)

Abstract

- The Dark Web is often misunderstood.
- The research paper explores its composition.
- Key Component: The Onion Router (TOR) browser.
  - Ensures Anonymity
  - Routes Traffic via Multiple Servers
- Difficult to control or monitor the dark web.
- Data Collection and Analysis
  - Using Data Mining & Penetration Testing Tools
- Ongoing Challenges
  - Progress in Crawling the Dark Web
  - Need for Continuous Adaptation

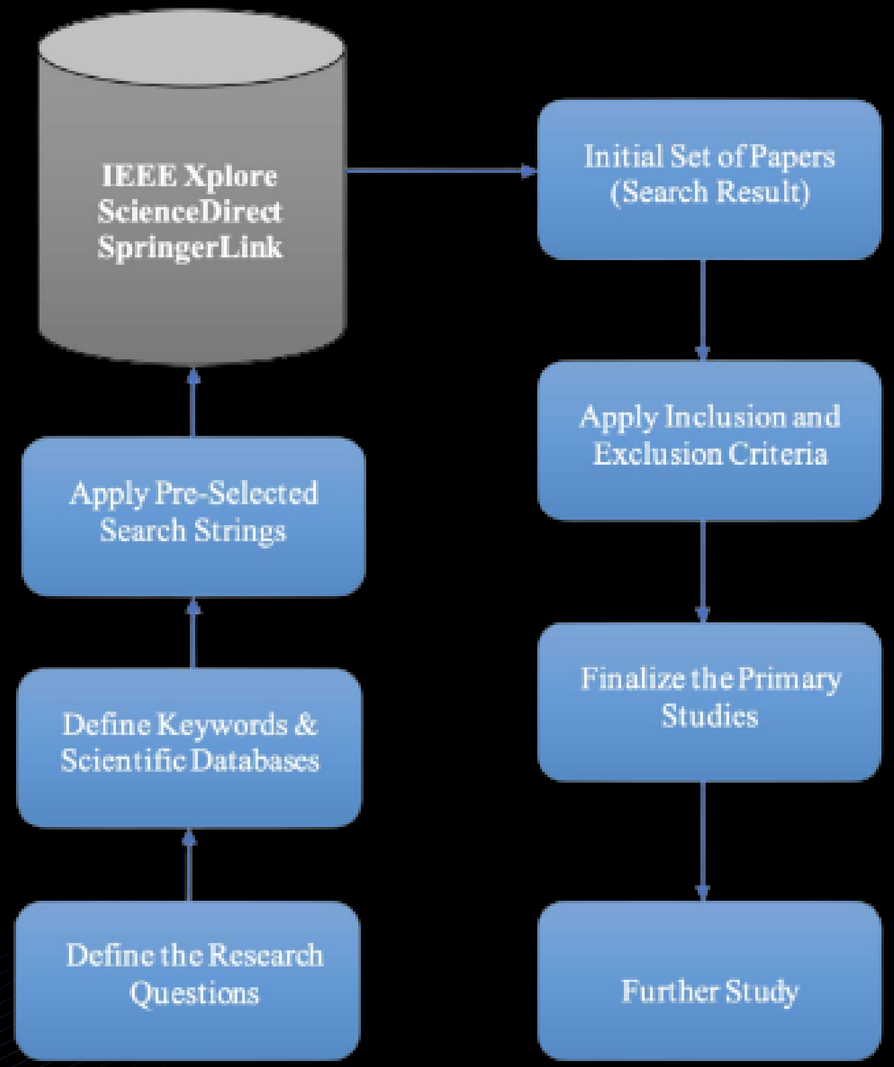


Introduction

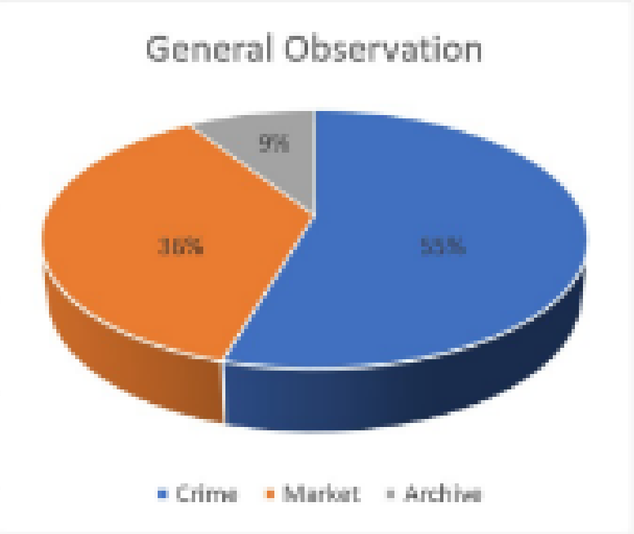
- Definition
  - Dark Web: Requires special software for access
  - Deep Web: Unreachable by normal browsers
- Scale
  - Dark Web: <5% of the internet
  - Deep Web: >90% of the internet
- Access Tool
  - The Onion Router (TOR): Ensures user anonymity
- Usage Insights
  - Bitcoin trade: $600 million (2017)
  - Silk Road: $1.2 billion (mid-2013)
- Research
  - Current state-of-the-art Dark Web technologies
  - Identifying remaining challenges
- Contributions
  - Systematic survey of Dark Web papers
  - Analysis of network security
  - Framework development

RESEARCH METHODOLOGY

1. In this section, the challenges and motivations for writing the paper are described.
2. A review of literature on Dark Web tech and ride-sharing was conducted.
3. A systematic review was undertaken using key databases.
4. Specific filters were applied, resulting in 196 studies.
5. Relevant papers were narrowed down through screening.

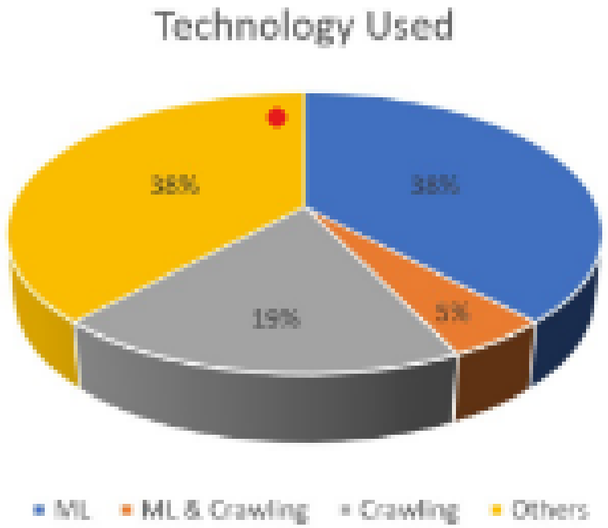


General Observation



Among **32 researched papers**, 11 i.e. **34%** of the papers were related to **general observation.**
In these observations, **55%** of the papers were related to **crime** and criminal activities in the dark web and **36%** were related to **dark web market** for **illegal activities**.

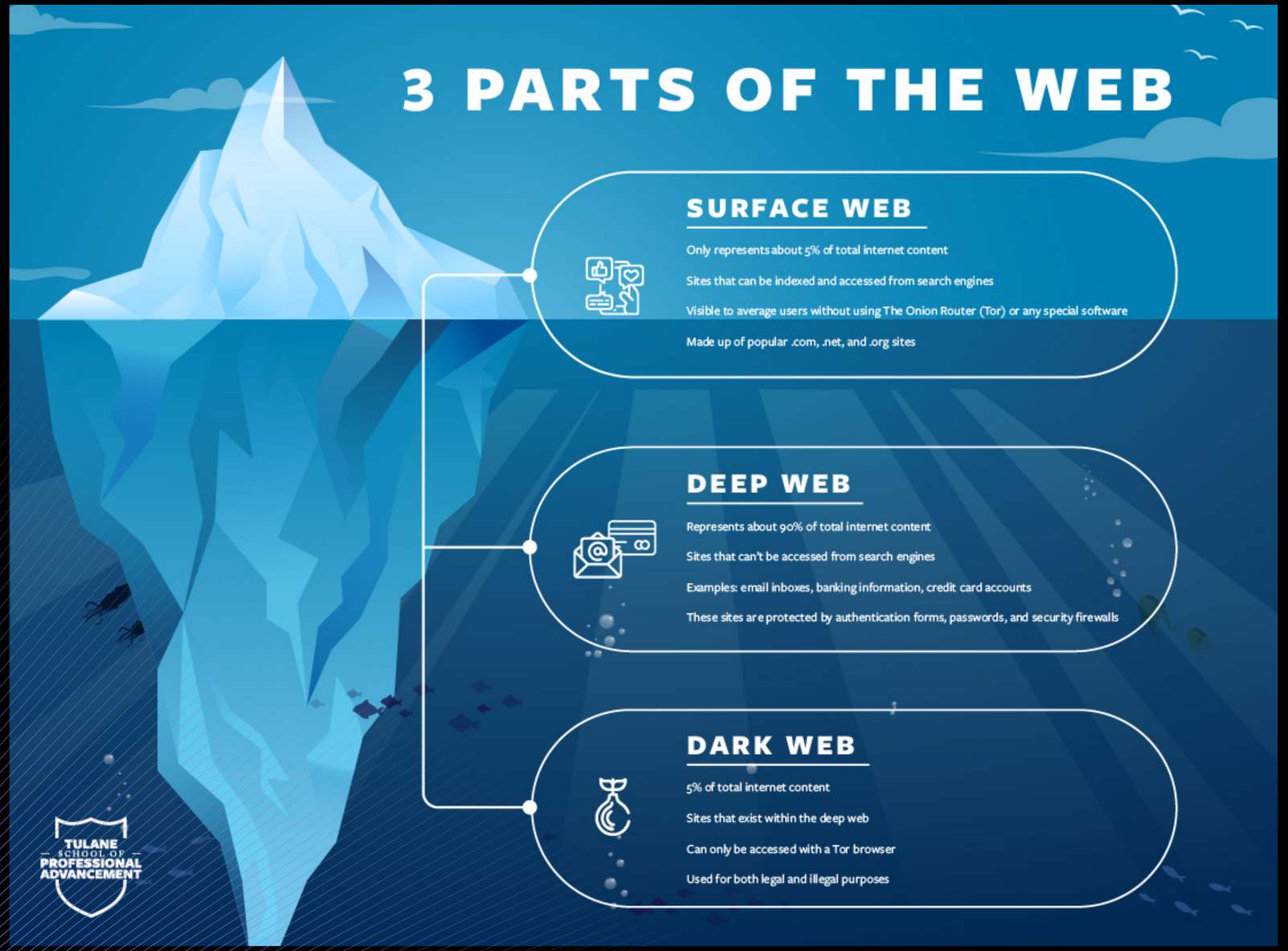
Different Technology



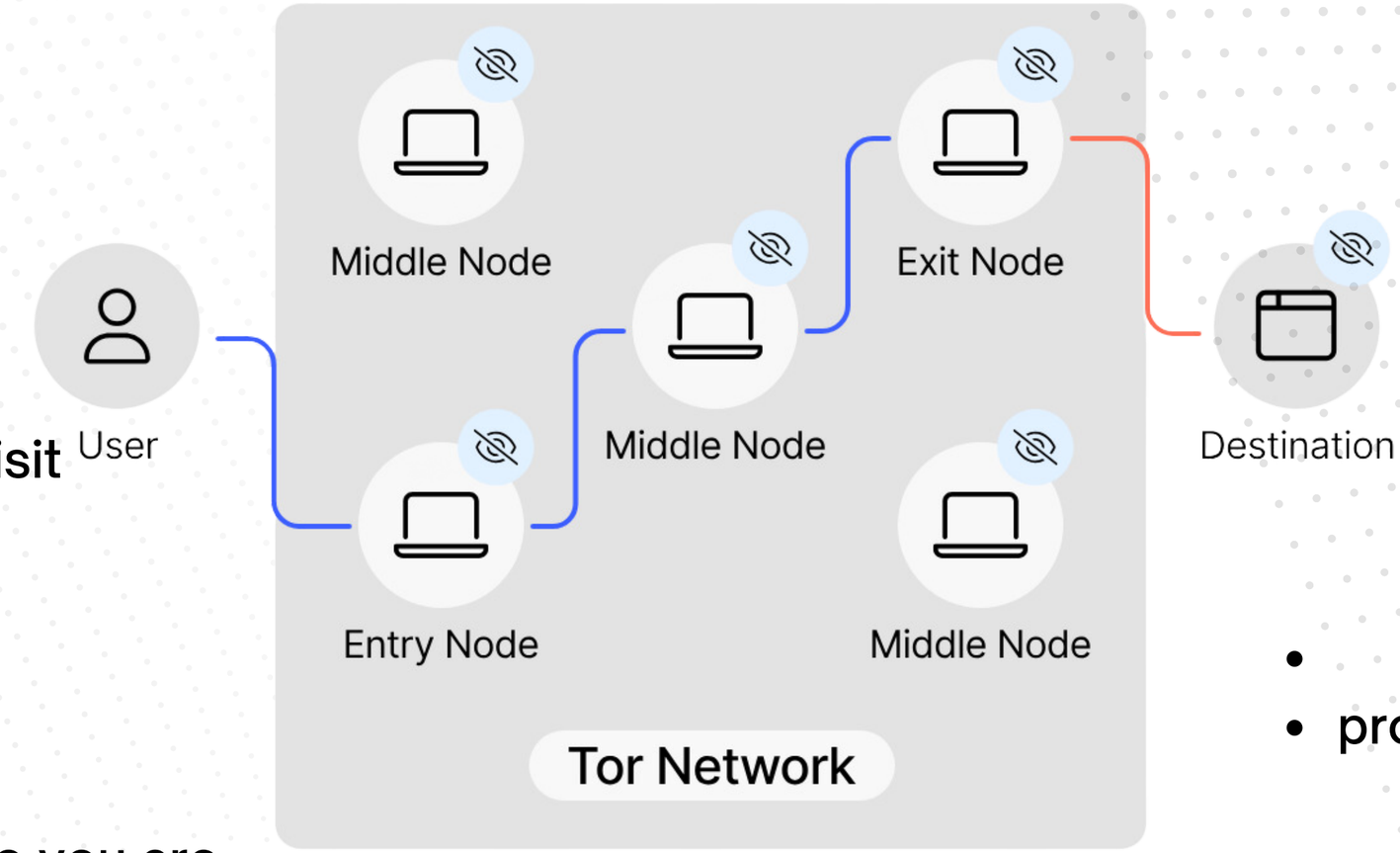
Among **32 researched papers**, 21 i.e. remaining **66%** of the papers were related to **different technology.**
In these papers, **38%** were related to **ML techniques** ,**19%** were related to **crawling techniques** and the rest **38%** were based on **techniques** related to devices used by **terrorists**.

What is Dark Web ?

- cannot be accessed by traditional browsers
- makes only up to 1% of the web
- virtually untraceable activities
- home to illegal maketplaces



Bins history
- browsing records disappear
- every search is a clean slate



Blocks trackers
- Isolates every site you visit
- ads cannot follow you

Encrypted traffic
- encrypts traffic
- protects anonymity

IP Masking
- hides real IP address
- websites don’t know who you are

Encrypted by Tor
Not Encrypted by Tor
IP address is hidden
IP address is known

General Observation
Papers

Basheer & Alkhtaib

Cyber Threat Intelligence

- major source of data and information
- still unable to analyze encrypted messages
- need of advanced tools

Godwatte

The Onion Router (TOR)

- vast and confusing road of internet
- host to biggest illegal marketplaces
- difficult to keeps tabs due to constantly changing URLs

Belshaw

Criminal Justice Programs

- increasing frequency on cyberattacks
- help shut down illegal schemes
- only 1.5% of users on TOR have access to hidden sites

Koch

Data Leaks

- 1 - attacks the flaws of tor software , are of technical level , most dangerous and rarely used
- 2 - exploits a computer's misconfigured settings to access sensitive information
- 3 - based on human error examples , providers forgetting racks or returning to a site multiple times

Technology Papers

Montieri

Anonymity Tools

- capable of hiding user’s identity towards the web server
- Traffic Classification helps in internet traffic engineering
- different algorithms are used

Chen

IED

- hard to find content pertaining to malicious activities
- crawling approach for discovery of IED

Marin

Hacking & Malware

- FireEye reported the exploit used to steal credit card information
- identifying the perpetrator may reveal the pattern, source and motive of the attack

Discussion

- Continued Exploration
  - Vast, evolving contents of the dark web.
  - Ongoing questions: Reliability of current marketplaces in 2022, evolving scraping methods, and machine learning advancements.
- Next Steps
  - Hands-on exploration of the Dark Web.
  - Developing a Web Crawler.
  - Attending conferences for deeper insight.
- Research Categories
  - Two main categories in reviewed papers:
    1. General observations on the dark web.
    2. Different technologies used.
- Technology Focus
  - Majority of papers focus on technology.
  - Key techniques: Machine Learning and Web Crawling.
  - Other techniques include Vector Space Model, Risk-based security, etc.
- Criminal Activities
  - Some papers highlight illegal activities in the dark web.
  - Focus on web archiving.
- Future Directions
  - Vast and complex nature of the dark web.
  - Ongoing research to crawl and understand it.
  - Potential to shut down illegal operations.

CONCLUSION

1.Cybersecurity Shield:
Dark web analysis serves as a shield for protecting systems and preserving online privacy.

2. TOR's Anonymity:
Specialized browsers like the Onion Router (TOR) empower users with anonymity, granting access to the hidden facets of the web.

3. Tool Evaluation:
We explored data mining and penetration testing tools, highlighting their strengths and limitations while unraveling the challenges in collecting dark web data.

4. Complexity Unveiled:
The dark web's intricate web of encrypted networks and obscured servers presents formidable obstacles to monitoring and control.

CONCLUSION

5. Future Research Horizons:
Promising research directions include automated fingerprinting and the evolution of tools like captcha breakers.

6. Wider Access:
Enhancements in captcha breakers are poised to broaden access and deepen investigations into the enigmatic dark web.

7. Cybersecurity Frontier:
The dark web continues to be a frontier in the realm of cybersecurity, demanding ongoing exploration and innovation.

8. Balancing Act:
As we advance, we must strike a balance between privacy and security, leveraging technology to navigate this hidden digital landscape.

Thank You!