Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ Altaf Alam

of IT Department, Semester V with

Roll No. 02 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

SECURITY LAB

LAB OUTCOMES

LO 1: Illustrate symmetric cryptography by implementing classical ciphers.

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

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

LO 4: Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.

LO 5: Use open-source tools to scan the network for vulnerabilities and simulate attacks.

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23

Experiment No 1

Aim : Breaking shift cipher and Mono Alphabetic Substitution cipher using Frequency analysis method.

Lab Outcome :

LO1: Illustrate symmetric cryptography by implementing classical ciphers.

Theory :

1. Shift Cipher:

Shift cipher, also known as Caesar cipher, is a type of substitution cipher where each letter in the plaintext is shifted a certain number of positions down the alphabet. It is one of the simplest and oldest encryption techniques.

Encryption Process:

- Each letter in the plaintext is replaced with the corresponding letter in the shifted alphabet. - The shift value 'k' determines how many positions each letter is moved. For example, with a shift of 3, 'A' becomes 'D', 'B' becomes 'E', and so on.
- Non-alphabetic characters, such as spaces or punctuation, are left unchanged in the ciphertext.
- The encryption formula is: $E(x) = (x + k) \bmod 26$, where 'x' is the numerical value of the letter and 'k' is the shift value.

Decryption Process:

- To decrypt the ciphertext, the receiver knows the shift value 'k' and simply shifts each letter backward in the alphabet. - The decryption formula is: $D(x) = (x - k) \bmod 26$.

Brute Force Attack on Shift Cipher:

Since there are only 25 possible shift values (excluding no shift or 0 shift), a brute force attack is feasible. The attacker can quickly try all combinations of shifts to decrypt the ciphertext and find the correct plaintext. Shift ciphers are not considered secure due to their vulnerability to brute force attacks.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23

2. Monoalphabetic Cipher:

A monoalphabetic cipher is a type of substitution cipher where each letter in the plaintext is replaced by the same corresponding letter in the ciphertext. The substitution remains constant throughout the encryption process.

Encryption Process:

- Each letter in the plaintext is replaced with a corresponding letter from the key. The key is a fixed 26-letter substitution table, where each letter in the alphabet is mapped to its respective substitution.
- For example, 'A' is replaced with the first letter of the key, 'B' with the second letter, and so on. - Non-alphabetic characters are left unchanged in the ciphertext.

Decryption Process:

- To decrypt the ciphertext, the receiver uses the same key to look up the corresponding plaintext letters for each letter in the ciphertext.

Brute Force Attack on Monoalphabetic Cipher:

A brute force attack on a monoalphabetic cipher is not practical because there are 26! (factorial) possible key combinations. This makes it computationally infeasible to try all combinations and decrypt the message.

Frequency Analysis Attack on Monoalphabetic Cipher:

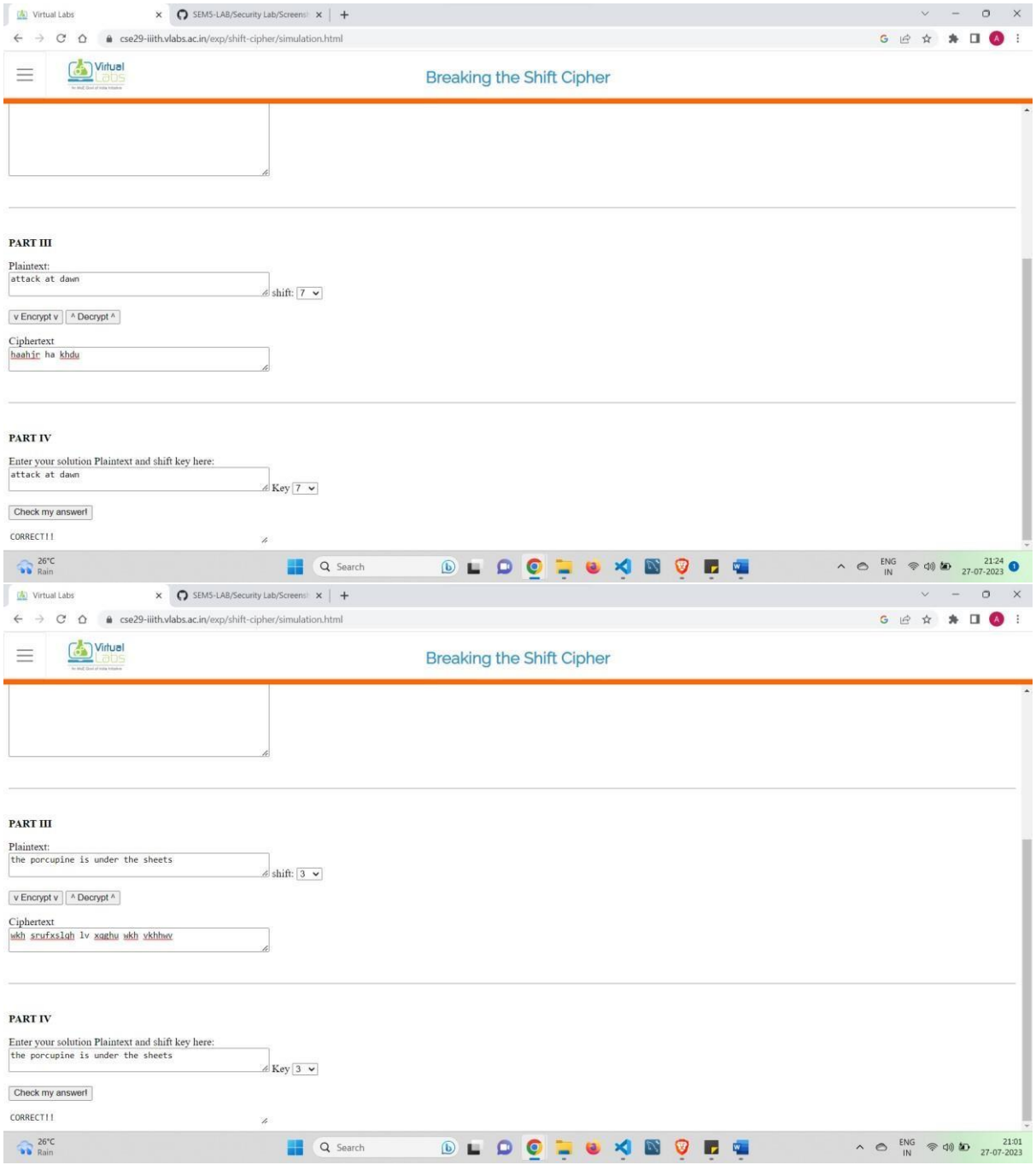
Frequency analysis is a powerful technique to break monoalphabetic ciphers. It exploits the fact that certain letters or groups of letters occur more frequently in the plaintext. For example, in English, the letter 'E' is the most common. By analyzing the frequency of letters in the ciphertext and comparing it with the expected frequency distribution in the English language, the attacker can deduce the key and decrypt the message.

Output:

1. Shift Cipher

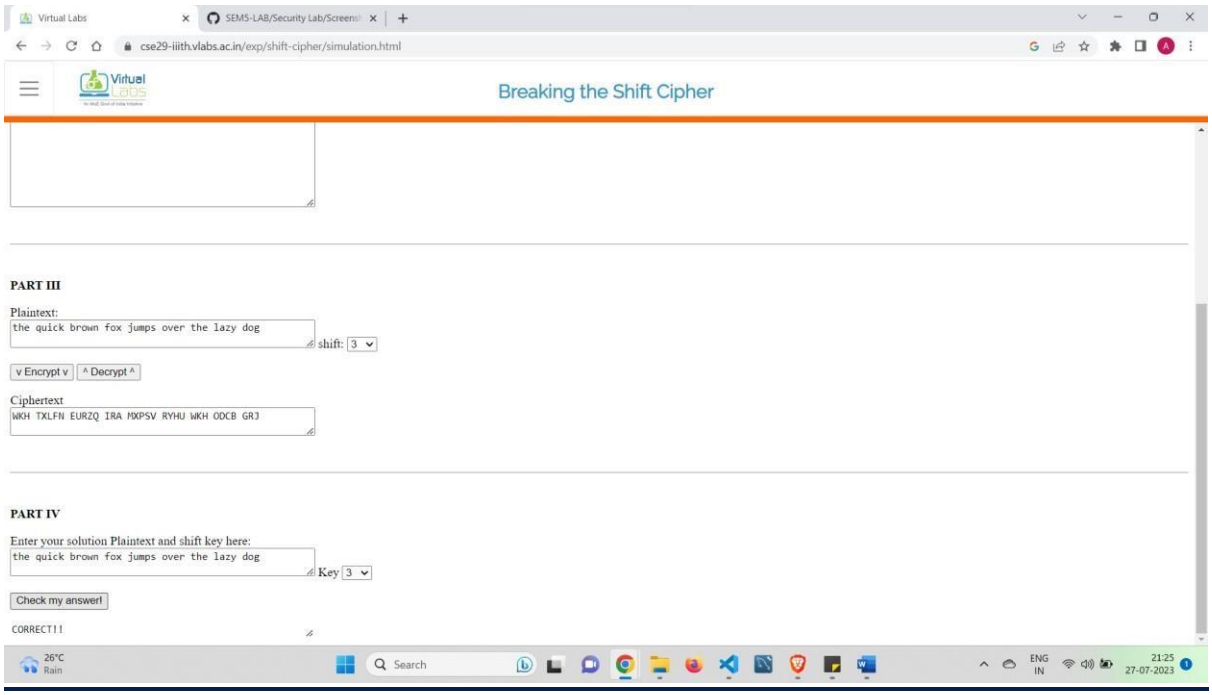
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23



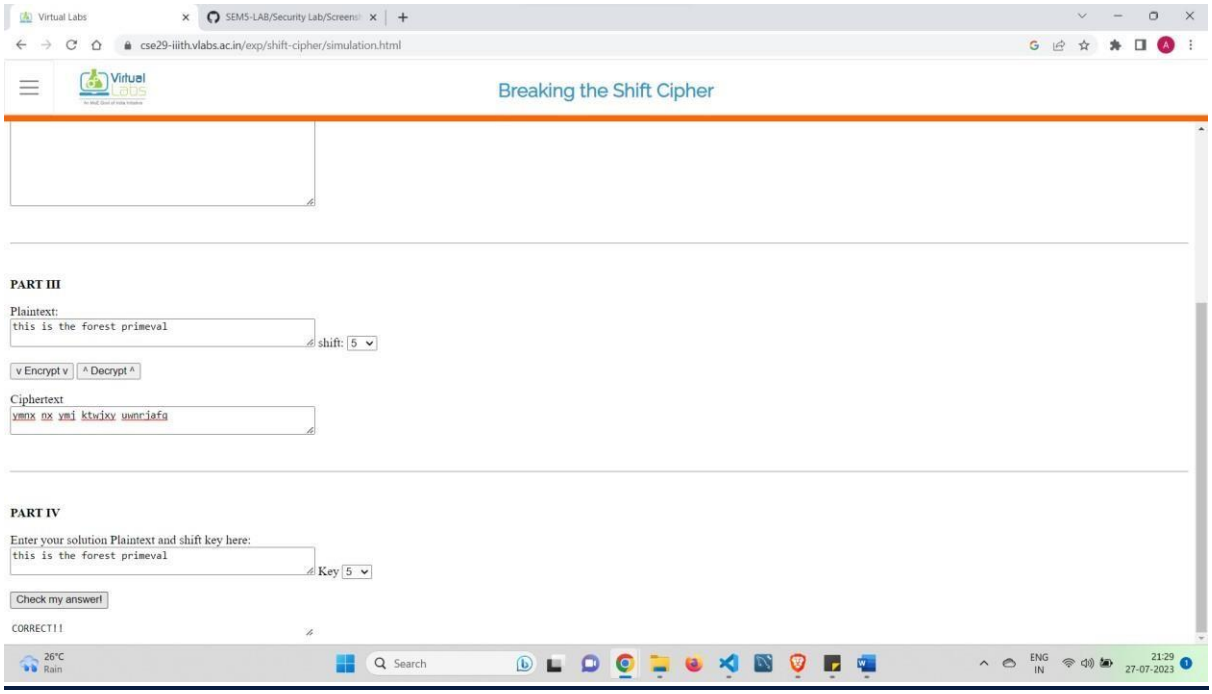
Name : Altaf Alam

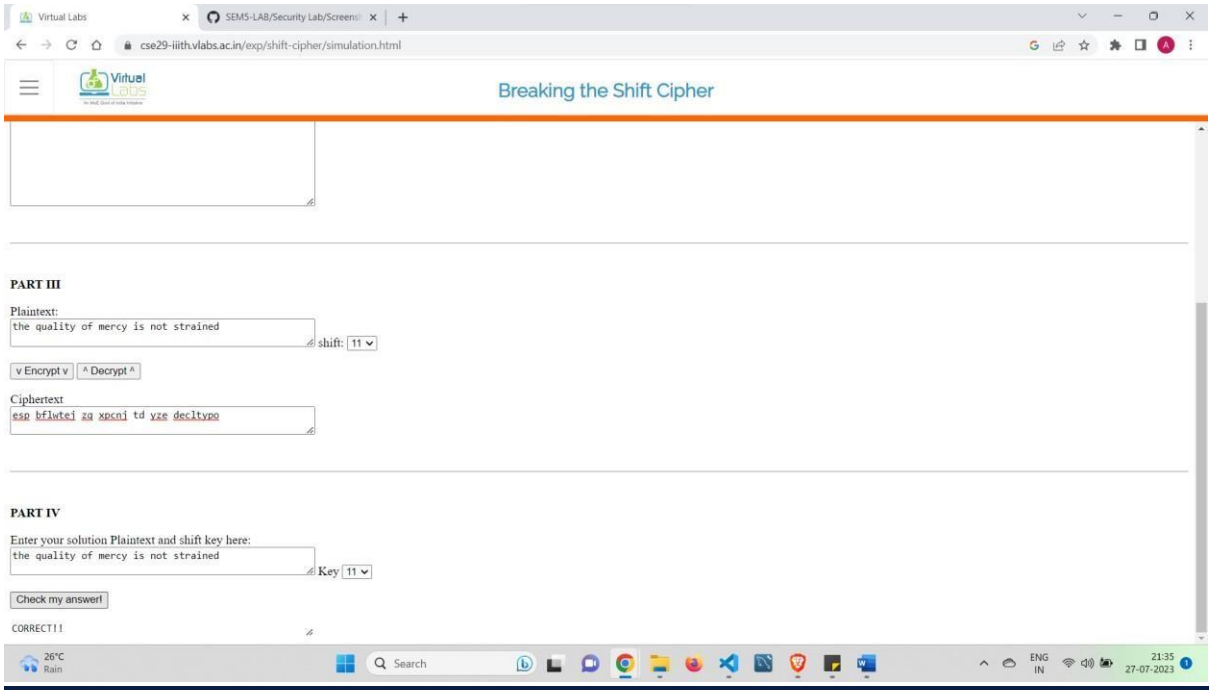
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23



Name : Altaf Alam

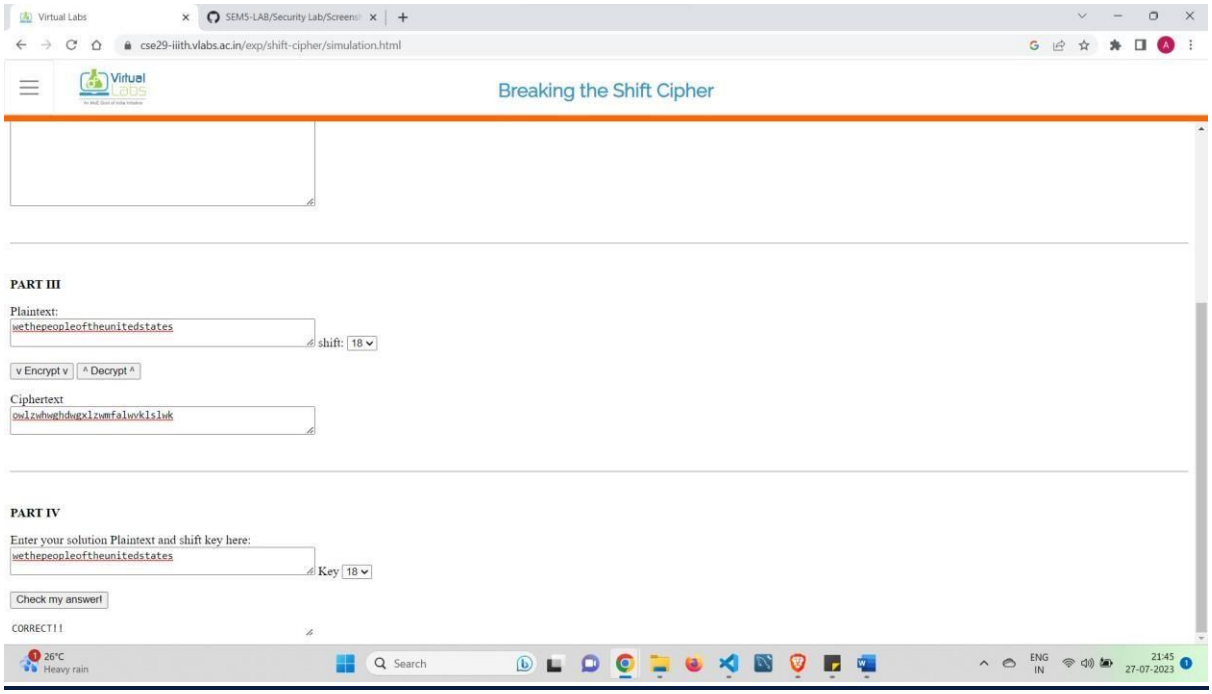
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23



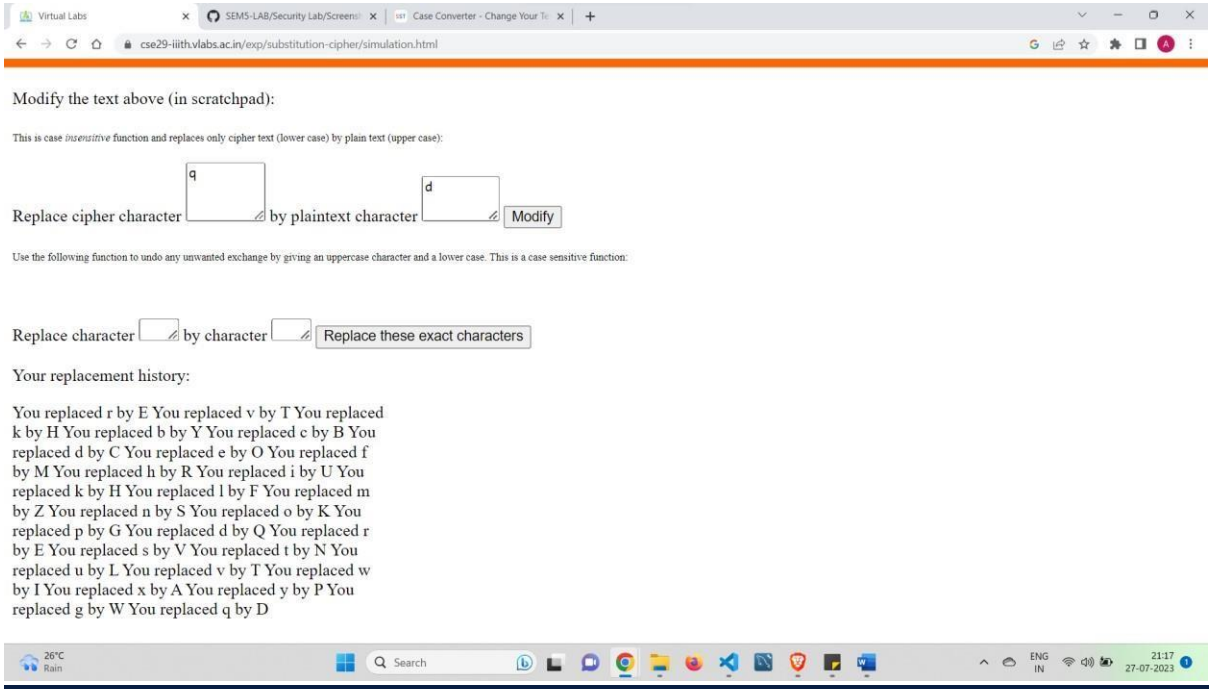


Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23

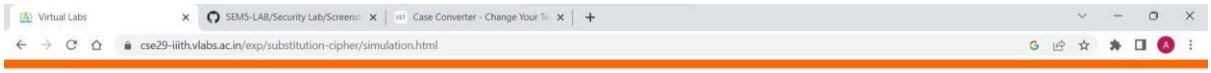


2. Substitution Cipher



Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23



PART III

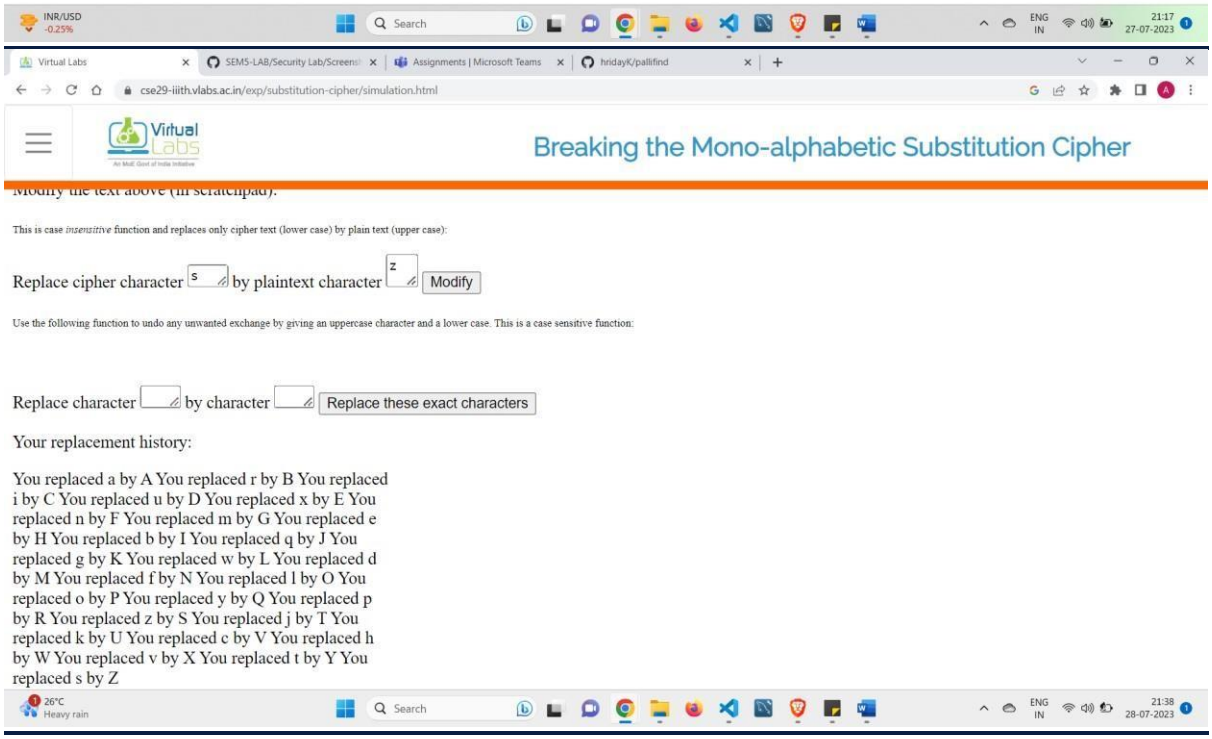
Enter your solution plaintext here:

```
CHAPTER 1 - DOWN THE RABBIT HOLE: ALICE IS BORED SITTING ON THE RIVERBANK WITH HER
SISTER, WHEN SHE NOTICES A TALKING, CLOTHED WHITE RABBIT WITH A POCKET WATCH RUN
PAST. SHE FOLLOWS IT DOWN A RABBIT HOLE WHEN SUDDENLY SHE FALLS A LONG WAY TO A
CURIOUS HALL WITH MANY LOCKED DOORS OF ALL SIZES. SHE FINDS A SMALL KEY TO A DOOR
TOO SMALL FOR HER TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN
DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK
TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO GROW TO SUCH A
TREMENDOUS SIZE HER HEAD HITS THE CEILING.
```

Solution Key = xcdqrlpkwzoufteyahnvisgjbm

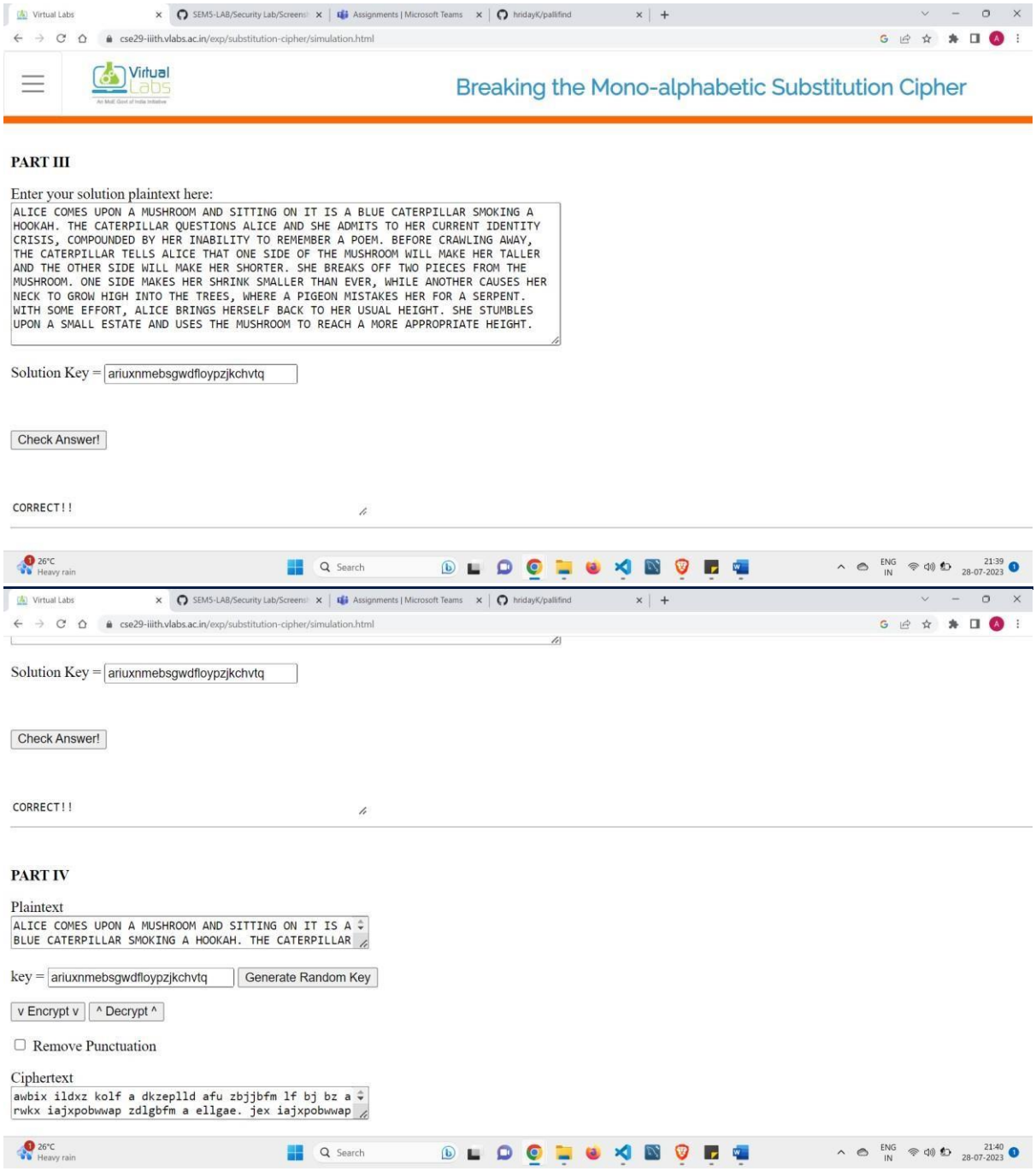
Check Answer!

CORRECT!!



Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 21/07/23



Conclusion :

Shift ciphers are simple and easy to implement, but they are not secure against brute force attacks due to the limited number of possible keys. On the other hand, monoalphabetic ciphers are more secure against brute force attacks due to the large number of potential keys, but they are vulnerable to frequency analysis attacks. To achieve stronger encryption, more complex encryption techniques like polyalphabetic ciphers or modern encryption algorithms should be used.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

Experiment No 2

Aim : Implementation and analysis of Cryptanalysis or decoding of polyalphabetic ciphers:

Playfair, Vigenere cipher.

Lab Outcome :

LO1 : Illustrate symmetric cryptography by implementing classical ciphers.

Theory :

1. How vigenere cipher works (with eg)?

The Vigenère cipher is a classical encryption technique that builds upon the simple Caesar cipher by introducing a keyword that determines the shifting pattern for each letter. This method of encryption offers a more robust form of security compared to the monoalphabetic substitution techniques.

How the Vigenère Cipher Works:

1. Choosing a Keyword:
Start by selecting a keyword, which can be any word or phrase. For instance, consider using the keyword "KEY" for this explanation.

2. Matching the Keyword Length:
Repeat the keyword to match the length of the plaintext you want to encrypt. If the plaintext is longer than the keyword, the keyword will be repeated as needed. Let's say we want to encrypt the word "HELLO."

3. Converting to Numerical Values:
Convert both the keyword and the plaintext into numerical values based on their positions in the alphabet. A=0, B=1, C=2, and so on. For our example, "HELLO" becomes 7 4 11 11 14.

4. Shifting with the Keyword:
Starting with the first letter of the keyword and the corresponding letter in the plaintext, add their numerical values together (mod 26) to get the encrypted letter. In our case:
- Letter "H" (numerical value 7) + Letter "K" (numerical value 10) = 17, which corresponds to the letter "R."
Continue this process for each letter, cycling through the keyword as necessary.

5. Converting Back to Letters:

Convert the numerical values of the encrypted letters back to their corresponding letters in the alphabet. The encrypted version of "HELLO" with the keyword "KEY" is "RIFNP."

Example:

- Plaintext: HELLO
- Keyword: KEY
- Numerical Values: 7 4 11 11 14
- Shifted Values: (7 + 10) (4 + 4) (11 + 24) (11 + 4) (14 + 10)
- Encrypted Values: 17 8 9 15 24
- Encrypted Text: RIOPI

Advantages and Weaknesses:

The Vigenère cipher's main advantage lies in its utilization of a keyword, introducing variability and making frequency analysis less effective. It can also handle more characters without becoming overly predictable. However, its security can still be compromised if the keyword is short or easily guessed.

2. Kasiski Test: Breaking the Vigenère Cipher:

The Kasiski Test is a cryptanalysis technique used to decipher the Vigenère cipher by determining the length of the keyword used for encryption. This method exploits the patterns that emerge when the same keyword segment encrypts multiple occurrences of the same plaintext segment, resulting in repeated ciphertext segments. By identifying the distances between these repetitions, cryptanalysts can deduce potential keyword lengths and subsequently break the encryption.

Steps of the Kasiski Test:

1. Identifying Repeated Ciphertext Segments:
Begin by examining the ciphertext for recurring segments. Due to the nature of the Vigenère cipher, if the same keyword segment is used to encrypt the same plaintext segment, the resulting ciphertext segments will match.

2. Calculating Distances:
Once you've identified the repeated ciphertext segments, calculate the distances between them in terms of the number of letters. These distances often correlate to multiples of the keyword length.

3. Finding Common Factors:

Look for common factors among the distances you've calculated. If the same keyword length was used multiple times, it would result in similar distances. By identifying common factors, you narrow down the possible lengths of the keyword.

4. Testing Potential Keyword Lengths:
With the potential keyword lengths in mind, you can begin testing them to see if they reveal any patterns. If a guessed keyword length is correct, it would result in a repeating pattern in the decrypted text, indicating that you're on the right track.

5. Deciphering the Text:
Once you've determined the keyword length, you can begin deciphering the text as if it's a set of simple Caesar ciphers. Divide the ciphertext into columns based on the keyword length, and decrypt each column using frequency analysis.

Example:

Let's say we have the following repeated segments in the ciphertext:
```
Ciphertext: GATKIVWLGGXKD
Segments: GAT KIVWLGG XKD
Distances: 4 7
```
The distances between the repeated segments are 4 and 7. The common factor between these distances is 1, suggesting that the keyword length could be 1 or a factor of the key length.

3. How Playfair Cipher works?

The Playfair cipher is a digraph substitution cipher that enhances security by encrypting pairs of letters rather than individual ones. This technique uses a key matrix, often a 5x5 grid, to transform plaintext letters into ciphertext digraphs. While it might seem intricate, understanding its operation can provide insights into its encryption process.

How Playfair Cipher Works:

1. Key Matrix Creation:
Begin by creating a key matrix, typically a 5x5 grid, containing unique letters. For example, consider the keyword "KEYWORD" and construct the matrix:

```
K E Y W O
R D A B C
F G H I L
M N P Q S
```

T U V X Z
```

2. Handling Repeated Letters and Missing Letters:
Since the key matrix requires distinct letters, some letters may be omitted, often a combination like "I" and "J." When encrypting, treat these as the same letter.

3. Breaking the Plaintext into Digraphs:
Divide the plaintext into pairs of letters (digraphs). For instance, "HELLO" becomes "HE" and "LL" with an added filler letter if needed. If the plaintext has an odd number of letters, append an extra letter (like "X") at the end.

4. Applying the Rules:
For each digraph:
- If both letters are in the same row, replace them with the letters to their right (looping to the leftmost if at the end).
- If both letters are in the same column, replace them with the letters below (looping to the top if at the bottom).
- If neither of the above conditions holds, form a rectangle with the two letters and replace each letter with the opposite corner's letter.

Example of Playfair Encryption:

- Key Matrix:
```
K E Y W O
R D A B C
F G H I L
M N P Q S
T U V X Z
```
- Plaintext: "HELLO" - Digraphs: "HE" and "LL" - Encryption Steps:
- "HE" forms a rectangle. Replace "H" with "L" and "E" with "O" to get "LO."
- "LL" is in the same row. Replace "L" with "M" and "L" with "N" to get "MN."
- Ciphertext: "LOMN"

Advantages and Limitations:

The Playfair cipher offers better security than simple substitution ciphers. Breaking it requires more complex techniques like frequency analysis of digraphs. The use of a key matrix introduces an additional layer of complexity, making cryptanalysis more intricate.
```

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

4. Cryptanalysis of the Playfair Cipher .

Cryptanalysis is the art of deciphering encrypted messages without the key. The Playfair cipher, a digraph substitution technique, may seem secure, but it's not immune to skilled cryptanalysts. Breaking the Playfair cipher involves exploiting its weaknesses and patterns to reveal the original message.

Cryptanalysis Steps for Playfair Cipher:

1. Frequency Analysis:
Even though Playfair encrypts digraphs, frequency analysis still works. In English, certain letter pairs are more common than others. Analyze the frequency distribution of digraphs in the ciphertext and compare them with known frequencies in English. High-frequency digraphs in the ciphertext could correspond to common letter pairs in the plaintext.

2. Identifying Patterns:
Look for repeated digraphs or sequences in the ciphertext. These can reveal underlying patterns that correspond to specific plaintext words or phrases. Patterns might emerge due to repeated sections of the original message.

3. Exploiting Known Plaintext:
If you have a portion of the original plaintext (a known plaintext), you can use it to your advantage. Encrypt the known plaintext with different parts of the key matrix to see if the ciphertext matches portions of the encrypted message. This might give you insights into the key matrix's structure.

4. Brute Force Attack:
If the key matrix is not very large or complex, a brute force attack might be feasible. Generate all possible key matrices and use each one to decrypt the ciphertext. Compare the decrypted results with English words or phrases to determine the correct key matrix.

5. Trial and Error with Keyword Variations:
If you have a hint about the keyword, try variations of the keyword to see if they yield meaningful plaintext. Small changes to the keyword can drastically alter the key matrix, affecting the decryption outcome.

6. Choosing an Optimal Key Length:
The length of the keyword affects the key matrix's size. If you can deduce the keyword length, you can narrow down the potential key matrices to test.

7. Breaking the Key Matrix:
If you have enough ciphertext, you might be able to identify repeated or near-repeated digraphs. This could indicate that the same key matrix sections are encrypting different parts of the

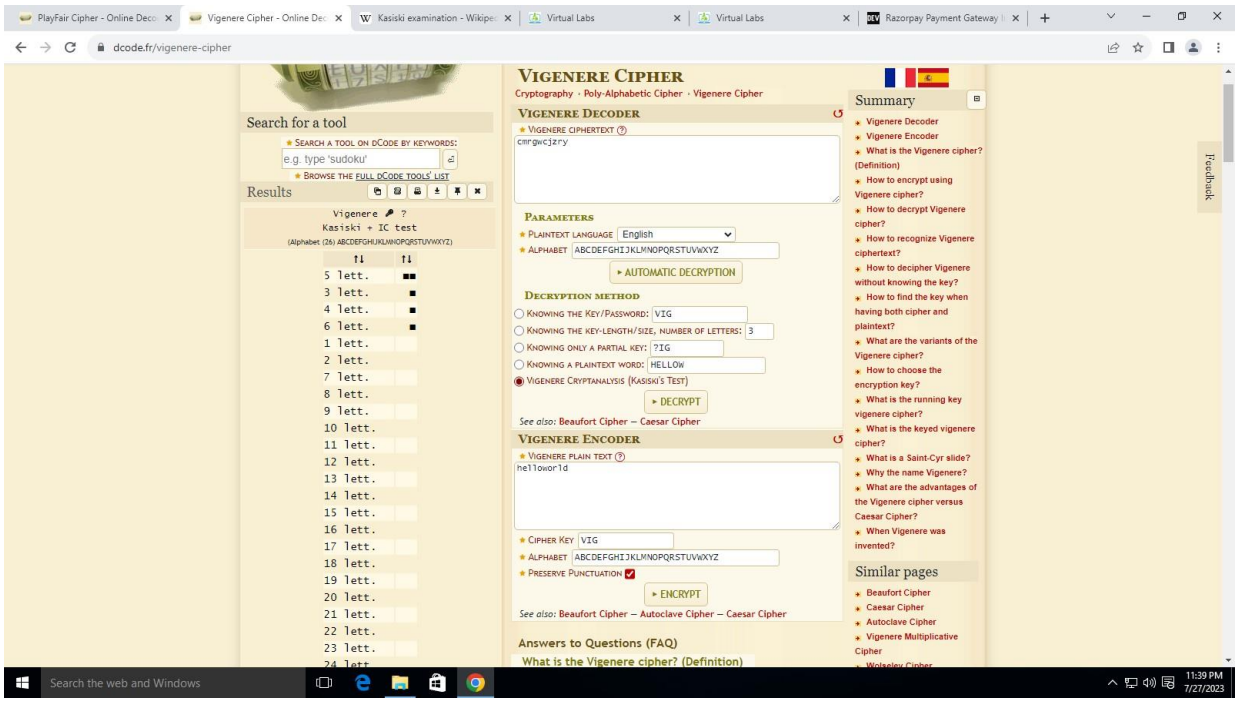
message. By analyzing these overlapping sections, you might be able to reverse-engineer parts of the key matrix.

Example:

Suppose you have the ciphertext "SRIOAKDLA." Analyzing the frequency of digraphs, you notice that "SR" is repeated. If "SR" corresponds to "TH" in English, you've made progress in deciphering the message.

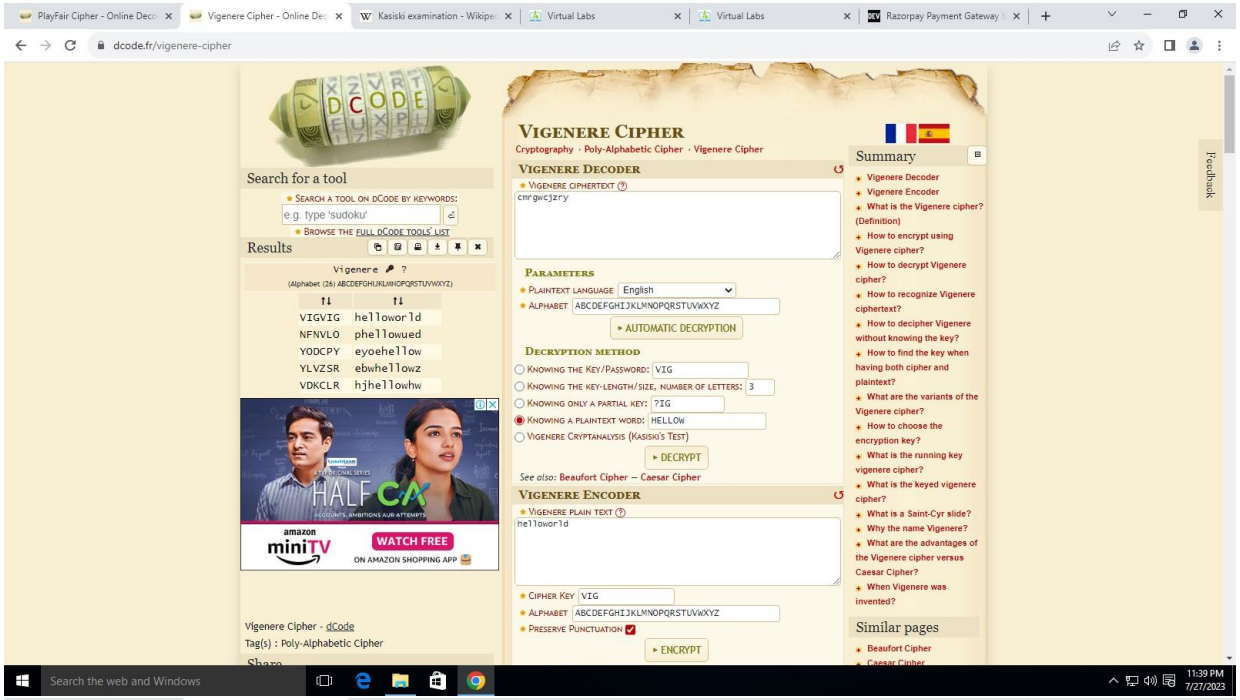
Output:

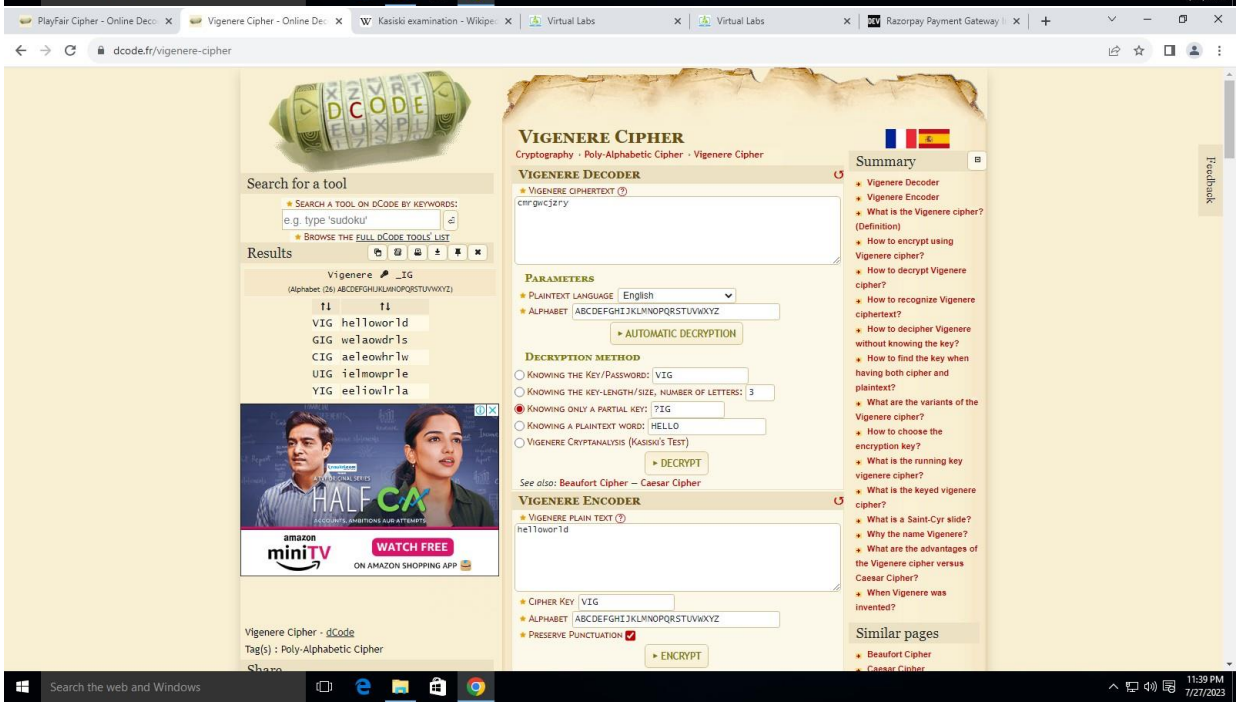
1. Vigenere Cipher:



Name : Altaf Alam

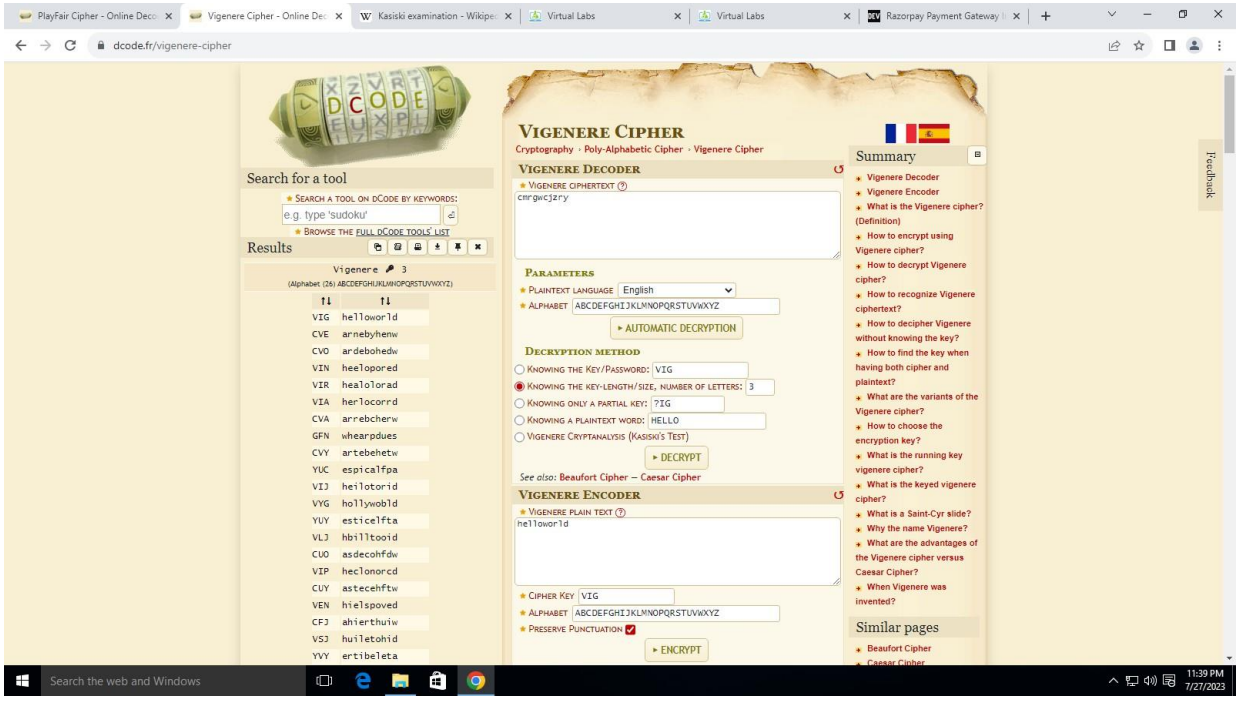
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

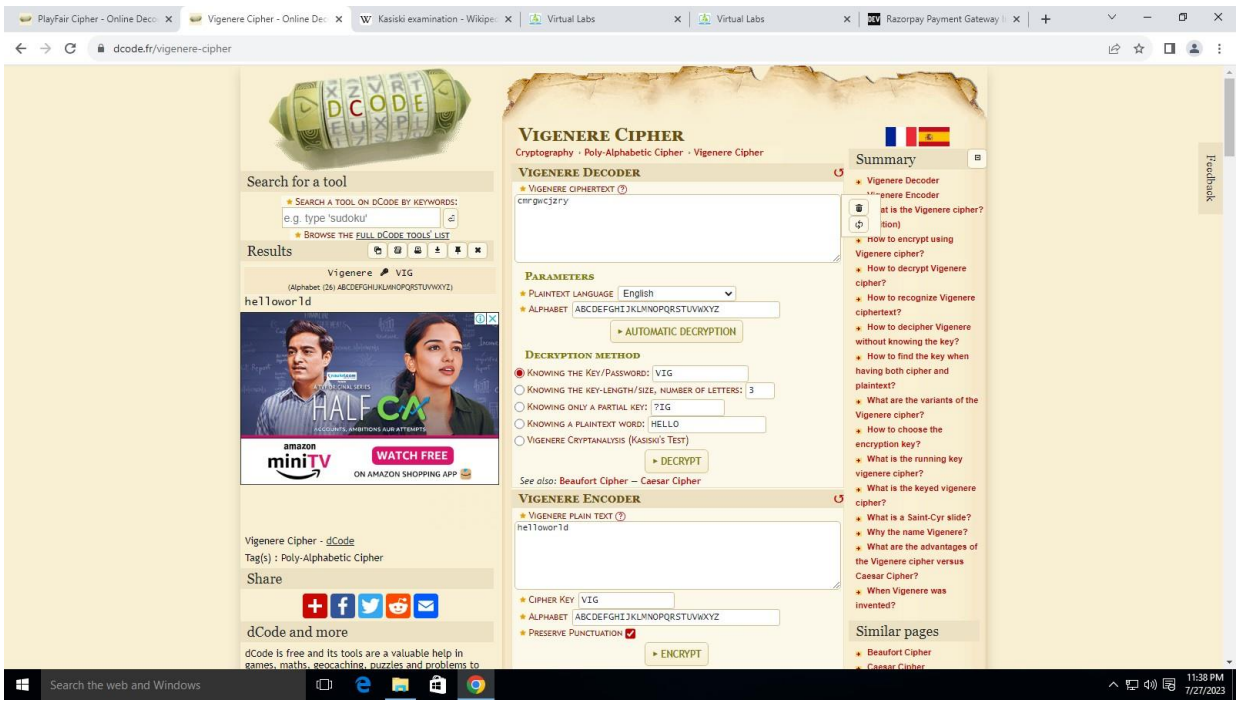




Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

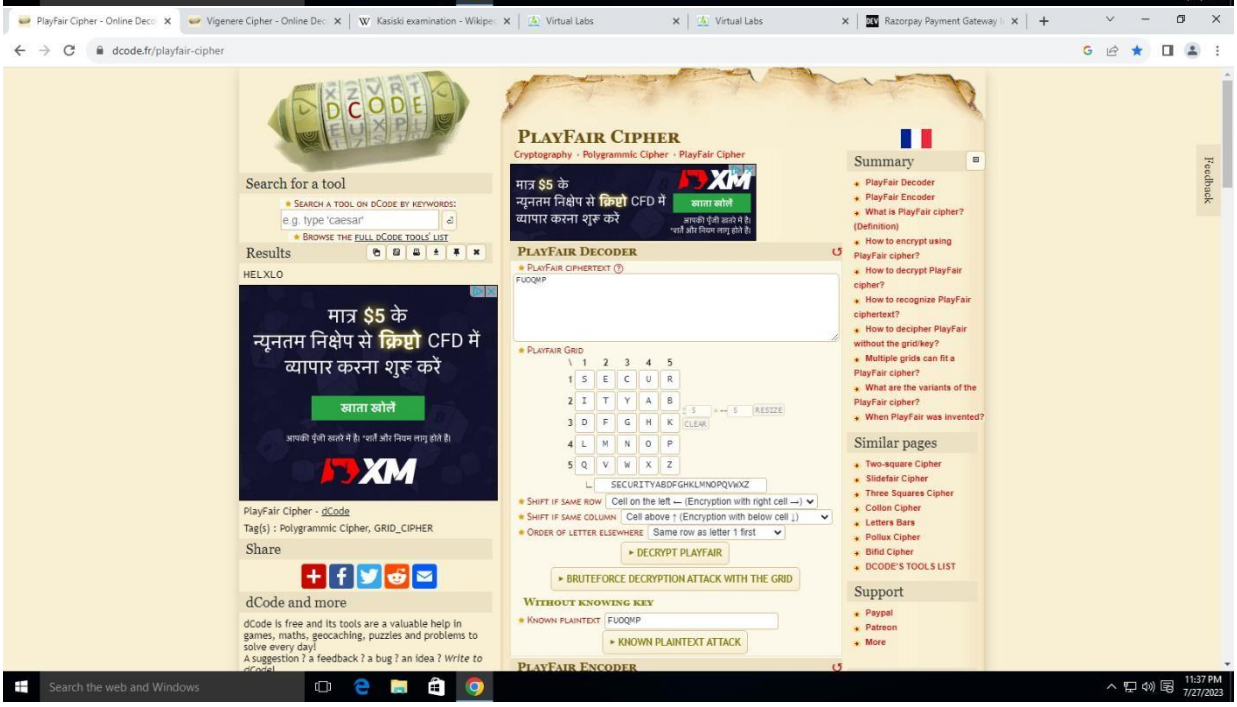




2. Playfair Cipher:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23



Conclusion:

Implemented and learned about Vigenère and Playfair Ciphers, explored the intricate steps of digital key generation and verification, successfully generated and verified digital signatures using software tools, and delved into the practical application of the RSA encryption scheme.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

Experiment No 3

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

Lab Outcome :

LO2

Theory :

1. AES Algorithm?

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

Cipher Type:

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

Number of Rounds:

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

- For AES-128: 10 rounds

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

Key Size:

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

Block Size:

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

Operations in Each Round:

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

1. SubBytes:

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

2. ShiftRows:

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

3. MixColumns:

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

4. AddRoundKey:

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

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

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

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

ECB mode

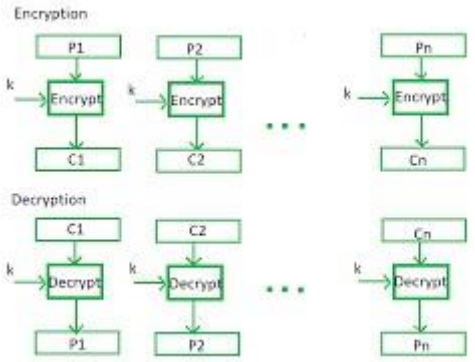
CBC mode

OFB mode

Counter mode

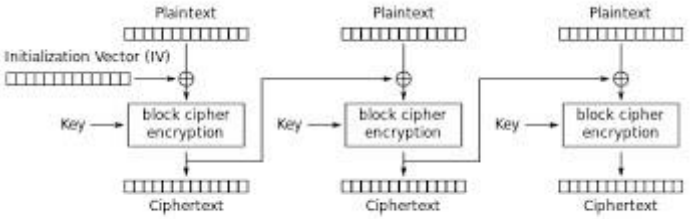
Block cipher modes of operation are techniques used to apply a block cipher, which is a cryptographic algorithm that encrypts fixed-size blocks of data, to larger amounts of data. These modes determine how blocks of plaintext are encrypted and how the resulting ciphertext is generated. Let's explore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

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



2. CBC (Cipher Block Chaining) Mode:

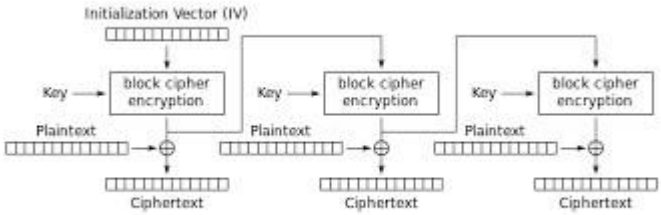
CBC mode addresses the weaknesses of ECB mode by introducing an Initialization Vector (IV) and chaining blocks together. Each plaintext block is XORed with the previous ciphertext block (or the IV for the first block), and then encrypted. This chaining introduces randomness and prevents identical blocks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



Cipher Block Chaining (CBC) mode encryption

3. OFB (Output Feedback) Mode:

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



Output Feedback (OFB) mode encryption

4. Counter Mode:

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

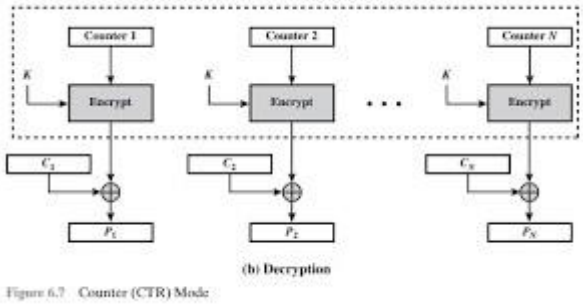


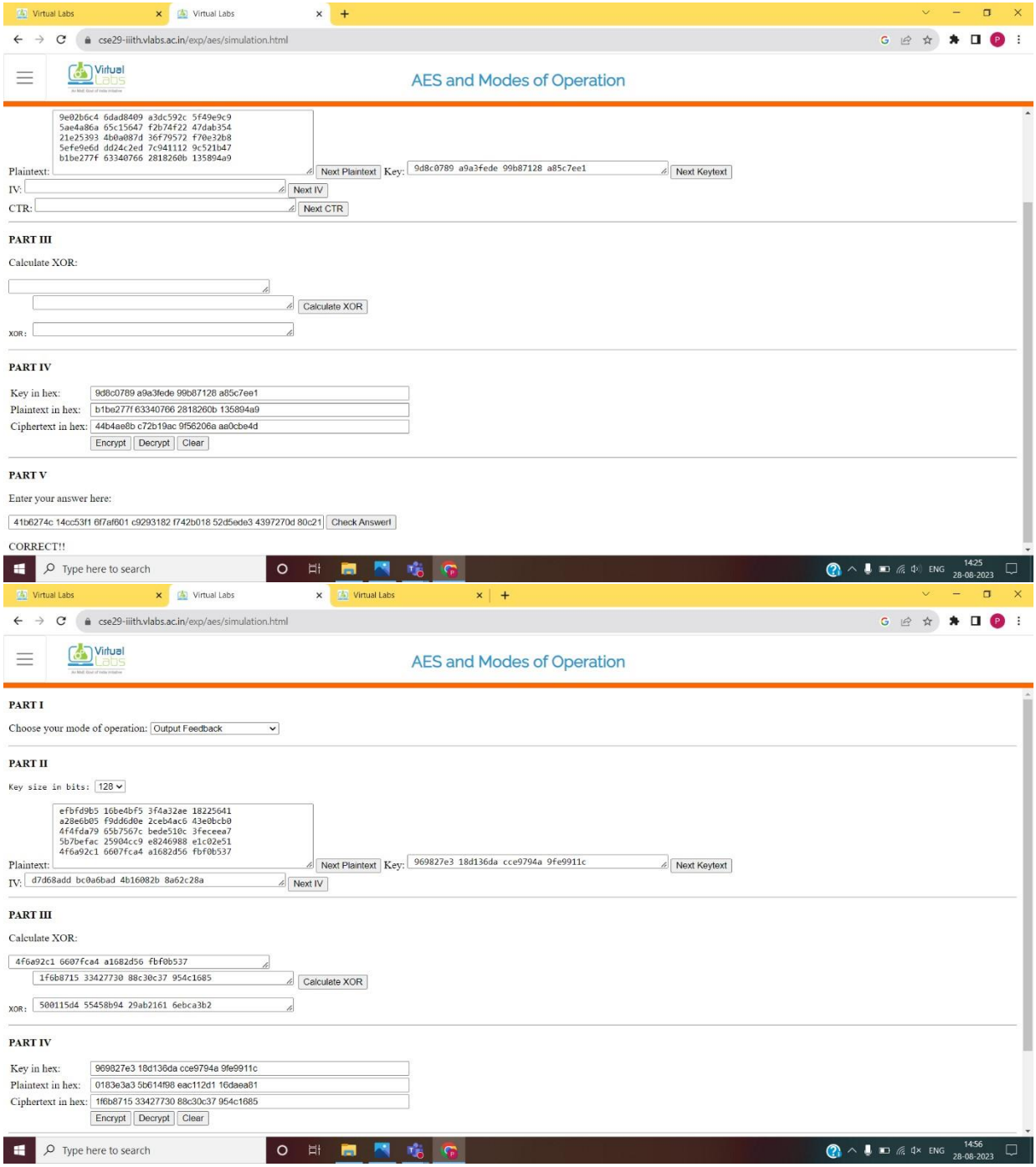
Figure 6.7 Counter (CTR) Mode

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

Output:

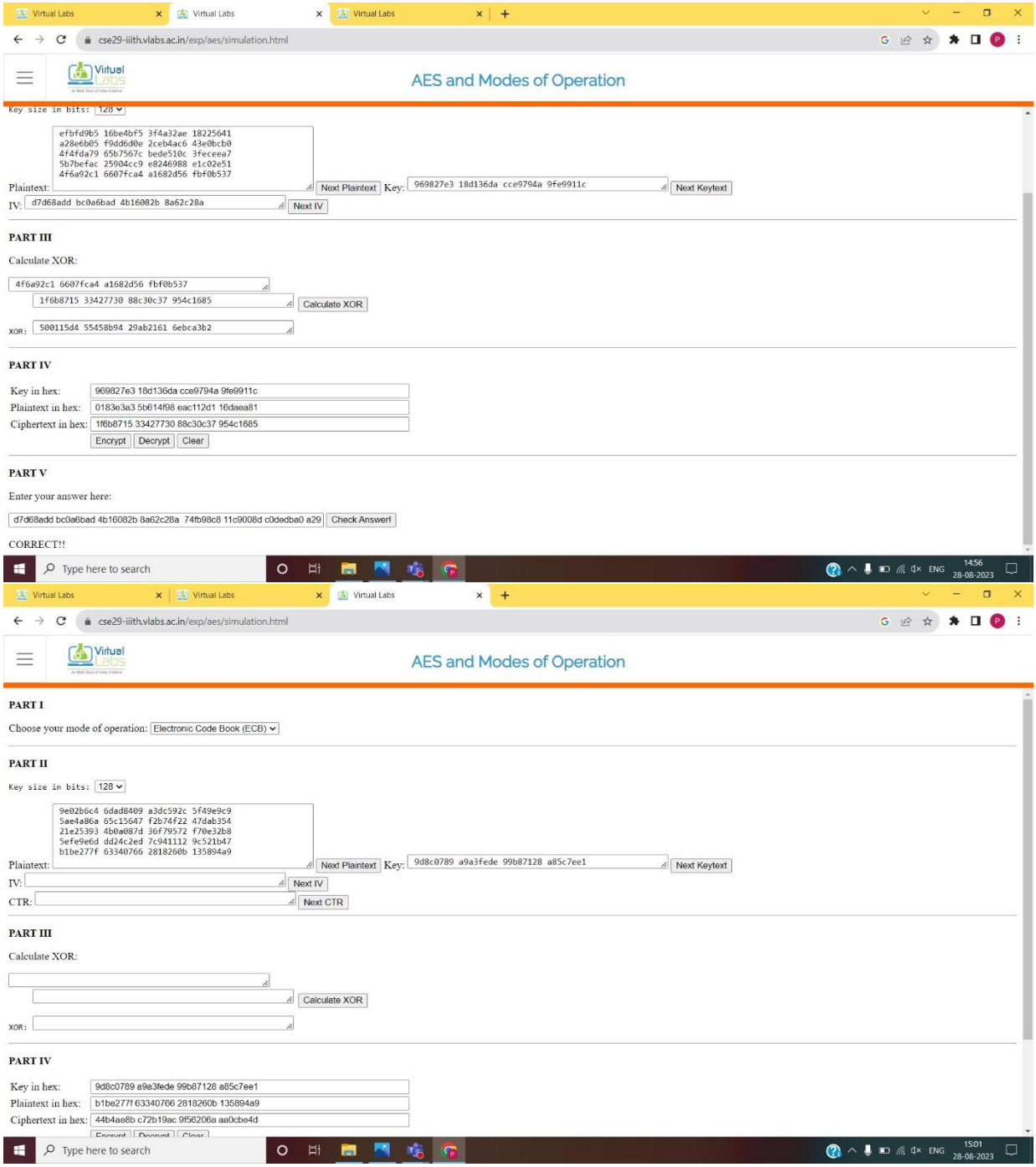
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



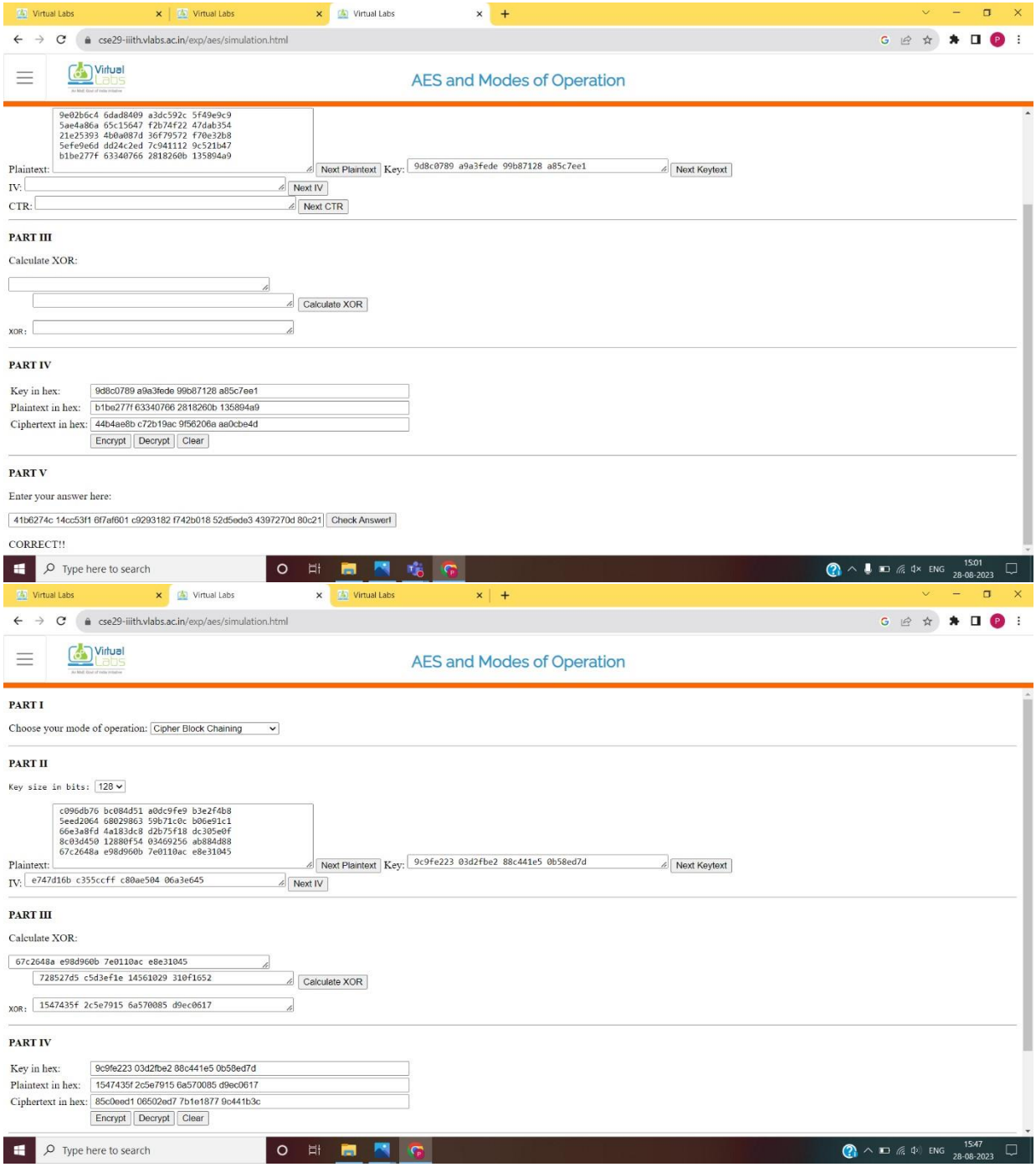
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



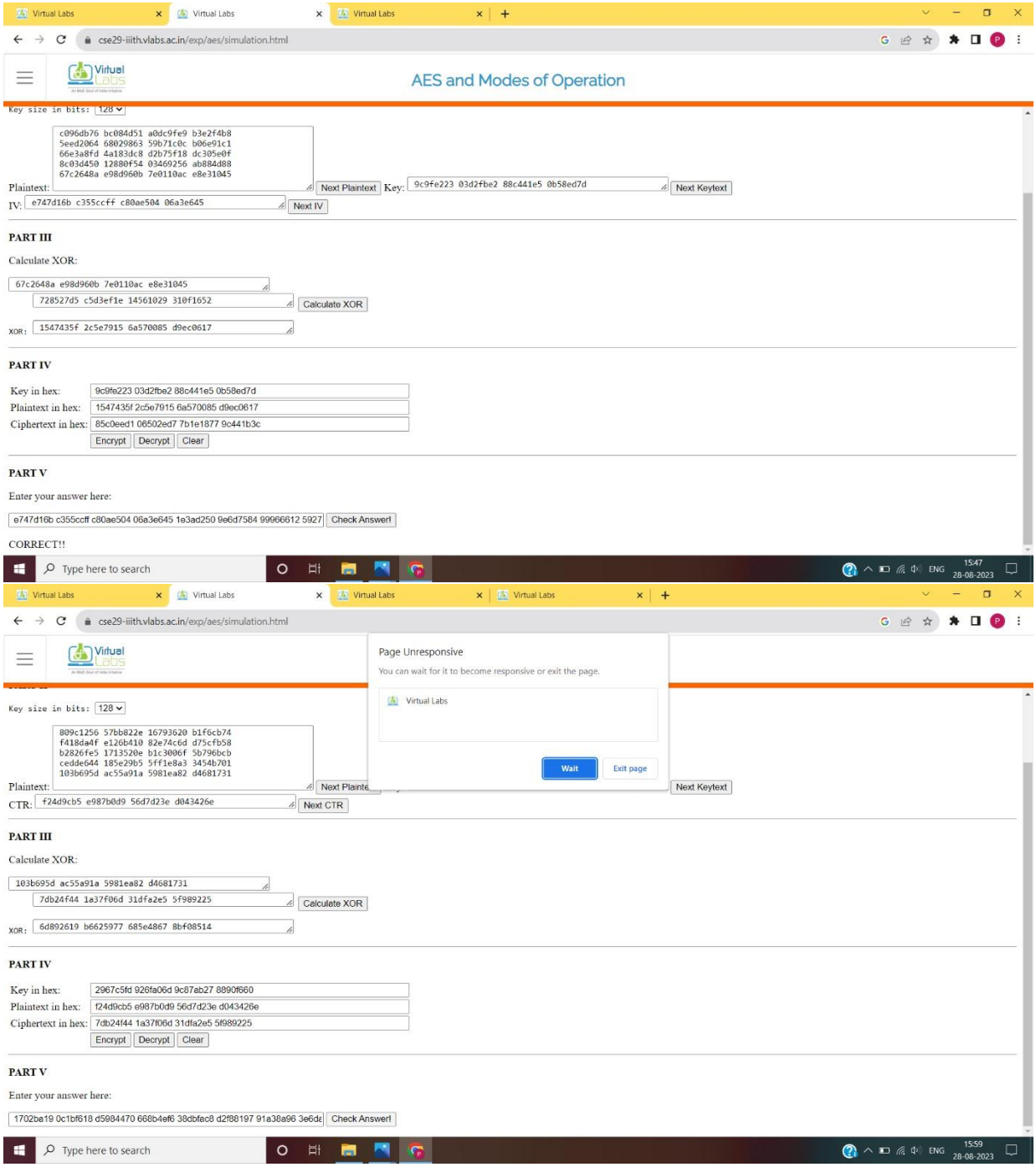
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



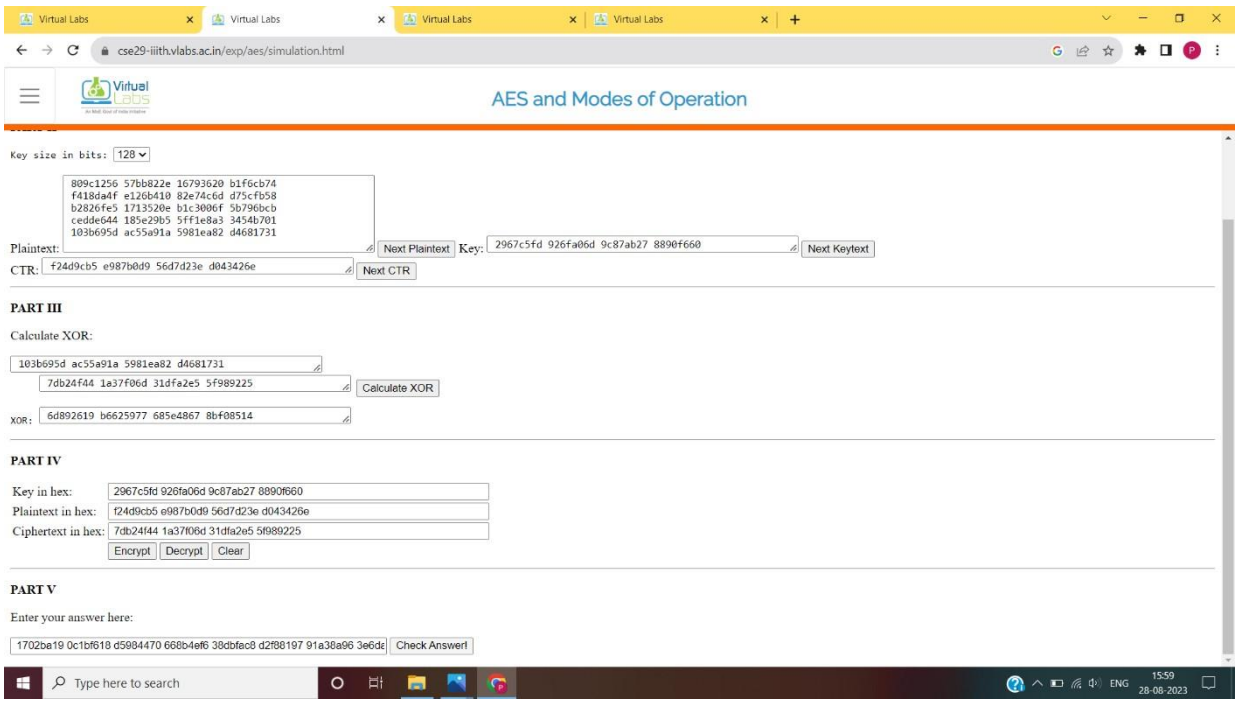
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



Conclusion:

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 08/08/23

Experiment No 4

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

Lab Outcome :

LO2 : Demonstrate Key management, distribution and user authentication.

Theory :

1. Explain the steps of RSA key generation?

RSA (Rivest-Shamir-Adleman) is a widely used asymmetric cryptographic algorithm that relies on a pair of keys: a public key and a private key. RSA key generation involves a series of steps that ensure secure communication and data encryption. Below, I'll provide a detailed explanation of the key generation process:

1. Choosing Prime Numbers (p and q): The first step in RSA key generation is to select two distinct prime numbers, usually denoted as p and q. These prime numbers are critical to the security of the algorithm. They need to be large and chosen randomly to prevent attackers from factoring the modulus and breaking the encryption.

2. Calculating Modulus (n): The modulus (n) is computed as the product of the two prime numbers: n = p * q. The modulus is used in both the public and private keys. It provides the size of the "space" in which the encryption operates, making the encryption stronger with larger values of n.

3. Calculating Euler's Totient Function (φ(n)): Euler's totient function (φ(n)) is calculated as φ(n) = (p - 1) * (q - 1). This function is crucial for ensuring that the public and private keys are relatively prime. φ(n) represents the count of numbers less than n that are coprime to n.

4. Selecting Public Exponent (e): The public exponent (e) is a small odd integer that is coprime to φ(n). A common choice for e is 65537 (2^16 + 1). This choice of e provides good security properties while ensuring efficient encryption and decryption operations.

5. Calculating Private Exponent (d): The private exponent (d) is computed as the modular multiplicative inverse of e modulo φ(n). In other words, it satisfies the equation (e * d) % φ(n) = 1. The private exponent d is what allows the decryption of messages encrypted with the public key.

6. Public Key Generation: The public key consists of the pair (e, n). It is distributed to anyone who wishes to send an encrypted message to the key owner. The public key is used for encrypting messages, ensuring only the private key holder can decrypt them.

7. Private Key Generation: The private key consists of the pair (d, n). This key must be kept secret and secure. The private key is used for decrypting messages encrypted with the corresponding public key. Losing the private key could compromise the security of encrypted communications.

The RSA key generation process is fundamental to the security and effectiveness of RSA encryption. It leverages the mathematical properties of prime numbers and modular arithmetic to create a secure communication channel between parties. The strength of RSA lies in the difficulty of factoring large semiprime numbers (the product of two large prime numbers), which makes it practically impossible for attackers to deduce the private key from the public key.

2. Explain the steps of Digital signature generation and verification process:

Digital signatures play a crucial role in ensuring data integrity, authenticity, and nonrepudiation in the digital world. They provide a way to verify that a digital document or message was indeed generated by a specific sender and has not been tampered with during transmission. The process involves two main steps: digital signature generation and digital signature verification.

Digital Signature Generation:

1. Hashing the Message: The sender begins by creating a cryptographic hash of the message they want to sign. A hash function, such as SHA-256, is used to produce a fixed-size digest that uniquely represents the content of the message.

2. Private Key Encryption: The sender then encrypts the hash value using their private key. This encrypted hash forms the digital signature. The use of the private key ensures that only the sender, who possesses the corresponding private key, can create the signature.

3. Attaching the Signature: The encrypted hash (digital signature) is attached to the original message. This combination forms the digitally signed message. Any alteration to the message will result in a different hash, making it evident that the message has been tampered with.

Digital Signature Verification:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 08/08/23

1. Hashing the Received Message: The recipient of the digitally signed message starts by computing the hash value of the received message using the same hash function as the sender. This generates a digest that represents the content of the received message.

2. Public Key Decryption: The recipient then decrypts the digital signature using the sender's public key. This process yields the original hash value that the sender encrypted using their private key during the signature generation.

3. Comparing Hashes: The recipient compares the decrypted hash value with the hash value they calculated from the received message. If the two hash values match, it indicates that the message has not been tampered with and that the signature is valid. A mismatch suggests either tampering or the use of an incorrect signature.

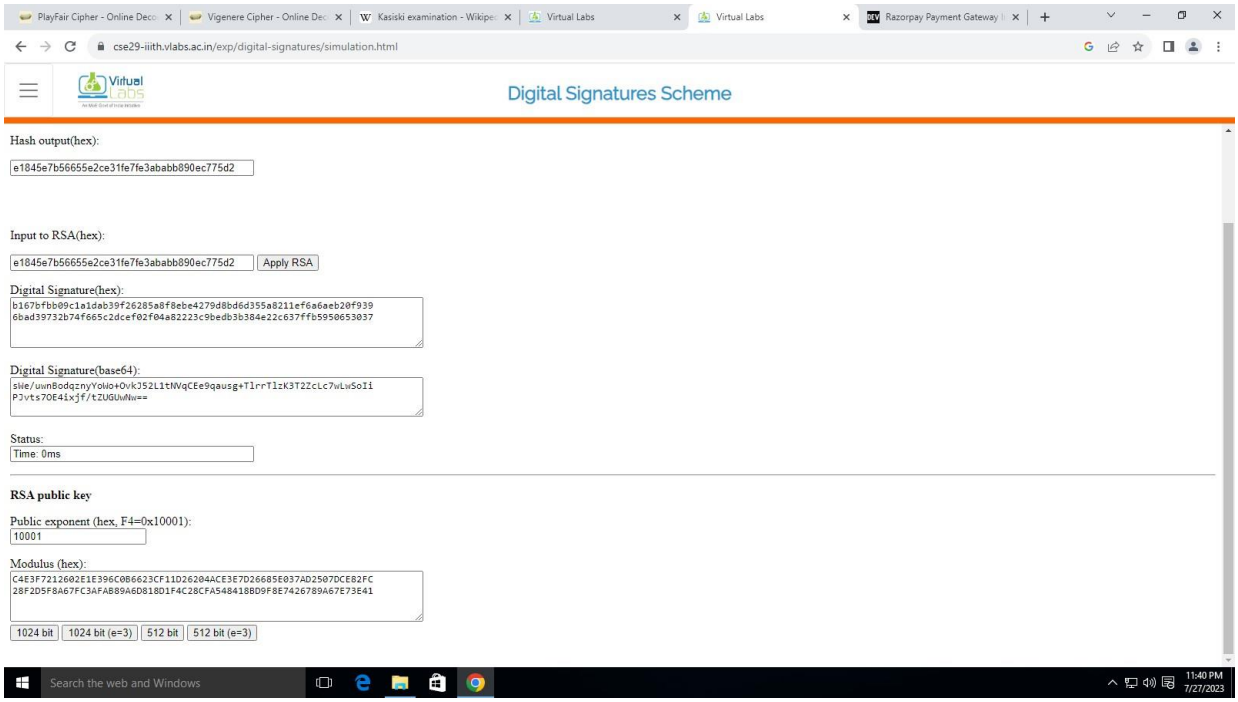
Benefits and Security:

The digital signature process provides several benefits:

- Data Integrity: Any alteration to the message or document will lead to a mismatch between the computed hash and the decrypted hash in the signature.
- Authenticity: The use of the sender's private key ensures that only the sender could have created the signature.
- Non-Repudiation: The sender cannot deny having sent the digitally signed message since the signature is tied to their private key.

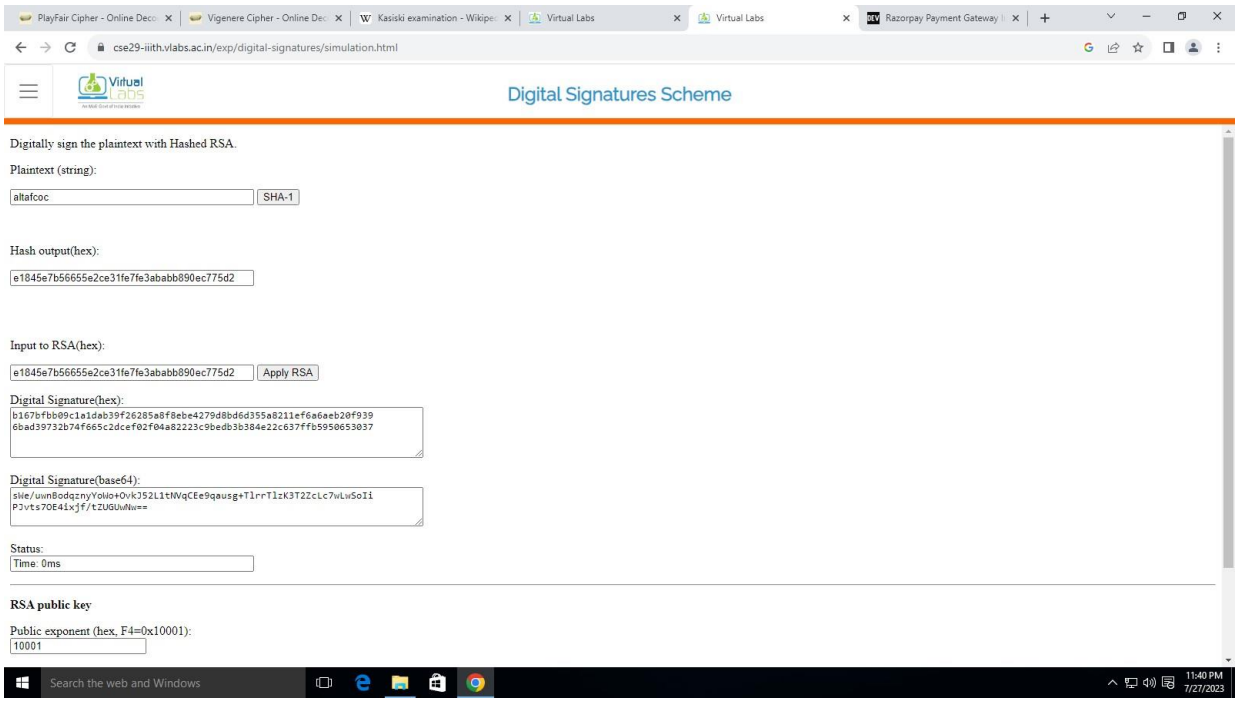
Output:

1. Digital Signature:



Name : Altaf Alam

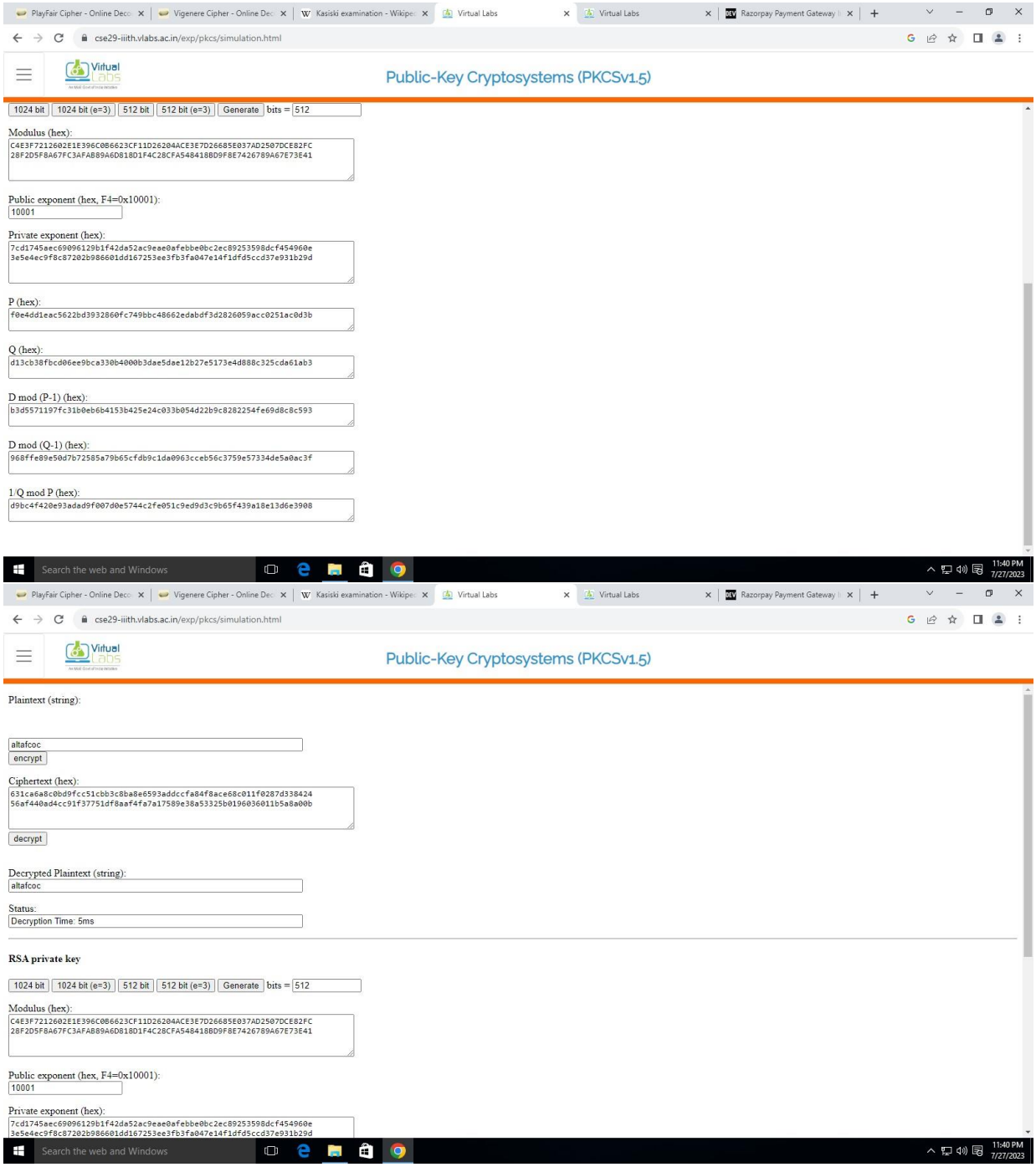
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 08/08/23



2. RSA key generation and decode

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 08/08/23



Conclusion:

Learnt about RSA scheme and RSA cryptosystem , explored steps involved in digital key generation and verification , generated and verified digital signature using software and also implemented RSA scheme.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

Experiment No 5

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

Lab Outcome :

LO2 :Demonstrate key management, distribution and user authentication.

Theory :

1. What is the need of hashing? List different hashing algorithms.

Hashing is a fundamental technique in computer science used to map data of arbitrary size to a fixed size. It plays a crucial role in various applications, such as data storage, encryption, and data retrieval. The primary purpose of hashing is to provide fast and efficient access to data, ensuring quick search, retrieval, and manipulation operations. Hashing also aids in ensuring data integrity and security. Let's explore the need for hashing in detail and discuss different hashing algorithms.

The Need for Hashing:

1. Efficient Data Retrieval: Hashing allows for quick data retrieval by generating a unique key for each data element. This key maps to a specific location in a data structure called a hash table, making lookups faster and more efficient. This is particularly useful in scenarios where large datasets need to be searched or accessed frequently.

2. Data Integrity and Security: Hashing is used in data integrity checks to ensure that the data hasn't been tampered with during transmission or storage. Hash functions generate a fixed-size hash value for a given input, and even a slight change in the input results in a completely different hash value. This property helps detect unauthorized modifications.

3. Cryptography and Security: Hashing is integral to cryptography, where it's used to secure passwords, digital signatures, and data encryption. Hash functions create a one-way transformation, making it computationally infeasible to reverse-engineer the original input from the hash value.

4. Distributed Systems: Hashing is used to distribute data across multiple servers in a distributed system. Each server's IP or identifier is hashed to determine where data should be stored. This helps balance the load and provides fault tolerance.

5. Caching: Hashing is employed in caching mechanisms to quickly determine whether a requested resource is already stored in the cache. This speeds up data retrieval and reduces the need to fetch data from slower sources.

6. Deduplication: Hashing is used to identify duplicate content or files. By comparing hash values, duplicate files can be identified and eliminated, saving storage space.

Different Hashing Algorithms:

1. MD5 (Message Digest Algorithm 5): Despite being widely used in the past, MD5 is considered cryptographically broken and unsuitable for further use due to its vulnerability to collision attacks. It produces a 128-bit hash value.

2. SHA-1 (Secure Hash Algorithm 1): Similar to MD5, SHA-1 is no longer recommended for secure applications due to its vulnerability to collision attacks. It produces a 160-bit hash value.

3. SHA-256 (Secure Hash Algorithm 256): Part of the SHA-2 family, SHA-256 produces a 256-bit hash value and is widely used for secure applications, including digital signatures and certificate authorities.

4. SHA-3: The latest member of the Secure Hash Algorithm family, SHA-3, was designed as a successor to SHA-2. It offers improved security and flexibility.

5. CRC32 (Cyclic Redundancy Check): CRC32 is used for error-checking and detecting changes to raw data. It's commonly used in network communications and storage systems.

6. MurmurHash: A non-cryptographic hash function known for its speed and distribution properties. It's often used in applications like hash tables, databases, and distributed systems.

7. CityHash: Another non-cryptographic hash function designed for hashing large amounts of data quickly. It's suitable for use in memory and storage systems.

8. Blake2: A cryptographic hash function that's faster than MD5, SHA-1, and even some of the SHA-3 variants. It's designed for security and high performance.

2. Write the commands used for generating hash values, matching them with stored hash values and auditing using hashdeep tool.

Hashdeep is a powerful command-line tool used for generating hash values, matching them with stored hash values, and performing auditing tasks on files and directories. It's widely used for verifying data integrity, ensuring files haven't been tampered with, and detecting changes in files. Here, we'll explain the commands for generating hash values, matching them, and conducting audits using hashdeep.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

1. Generating Hash Values:
To generate hash values using hashdeep, you can use the `-r` option to recursively hash files within a directory. The most commonly used hash algorithms include MD5, SHA-1, SHA-256, and others.

- Generate MD5 hash values for all files in a directory:
```
hashdeep -r -a md5 directory_path > hash.txt
```
- Generate SHA-256 hash values for all files in a directory:
```
hashdeep -r -a sha256 directory_path > hash.txt
```
2. Matching Hash Values:
Matching hash values involves comparing previously generated hash values with current ones to ensure file integrity. You can use the `-k` option to match hash values from a hash file.

- Match hash values from a hash file (generated using hashdeep) with current files in a directory:
```
hashdeep -r -k hash.txt directory_path
```
3. Auditing:
Auditing with hashdeep is used to compare hash values stored in hash files with files on the system to detect changes or discrepancies.

- Audit files in a directory against hash values in a hash file:
```
hashdeep -r -k -a -v hash.txt directory_path
```
Important Flags:
- `-r`: Recursively process files within directories.
- `-a <algorithm>`: Specify the hash algorithm (e.g., md5, sha256).
- `-k`: Use hash values from a hash file for matching.
- `-v`: Display verbose output, showing file names and hash value matches.
- `-c`: Enable strict mode to exit with an error code if any mismatches are found.

Example Scenario:
Suppose you have a directory named `data` containing files, and you want to generate SHA-256 hash values for all files, store them in a hash file named `hash.txt`, and then verify their integrity:

1. Generating hash values:
```
hashdeep -r -a sha256 data > hash.txt
```

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

```
```

2. Matching hash values:

```
hashdeep -r -k hash.txt data
```

3. Auditing:

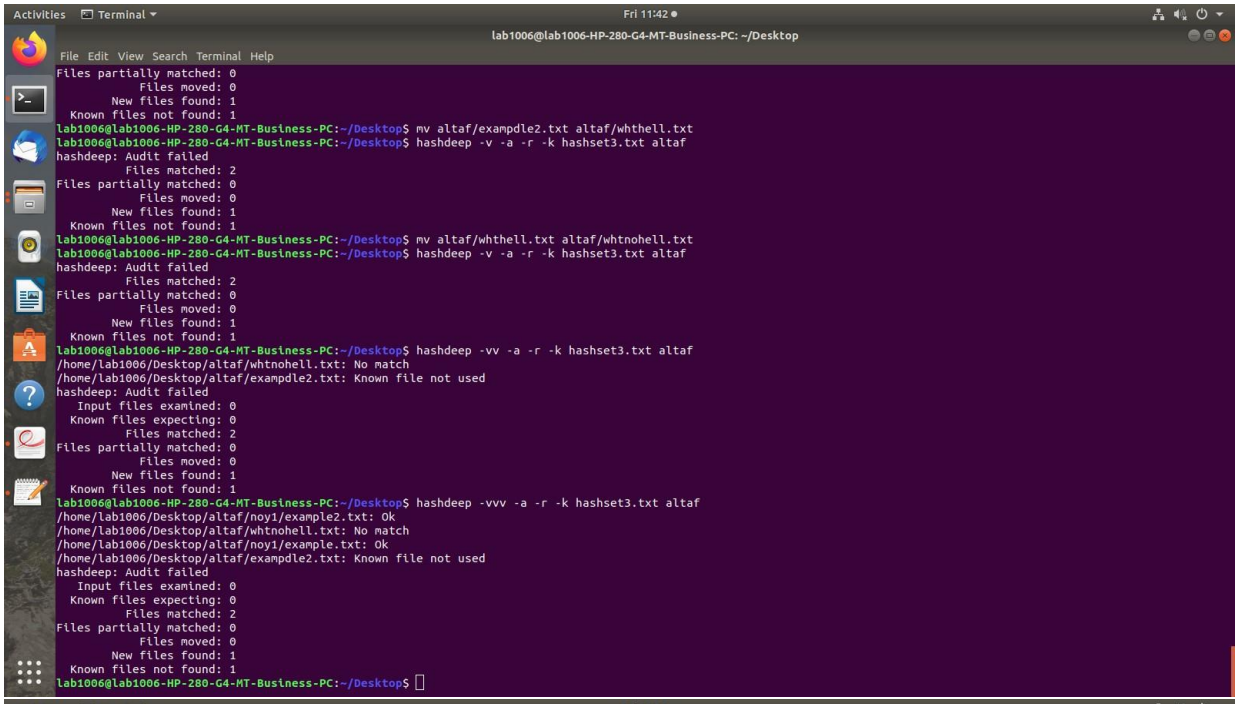
```
hashdeep -r -k -a -v hash.txt data
```

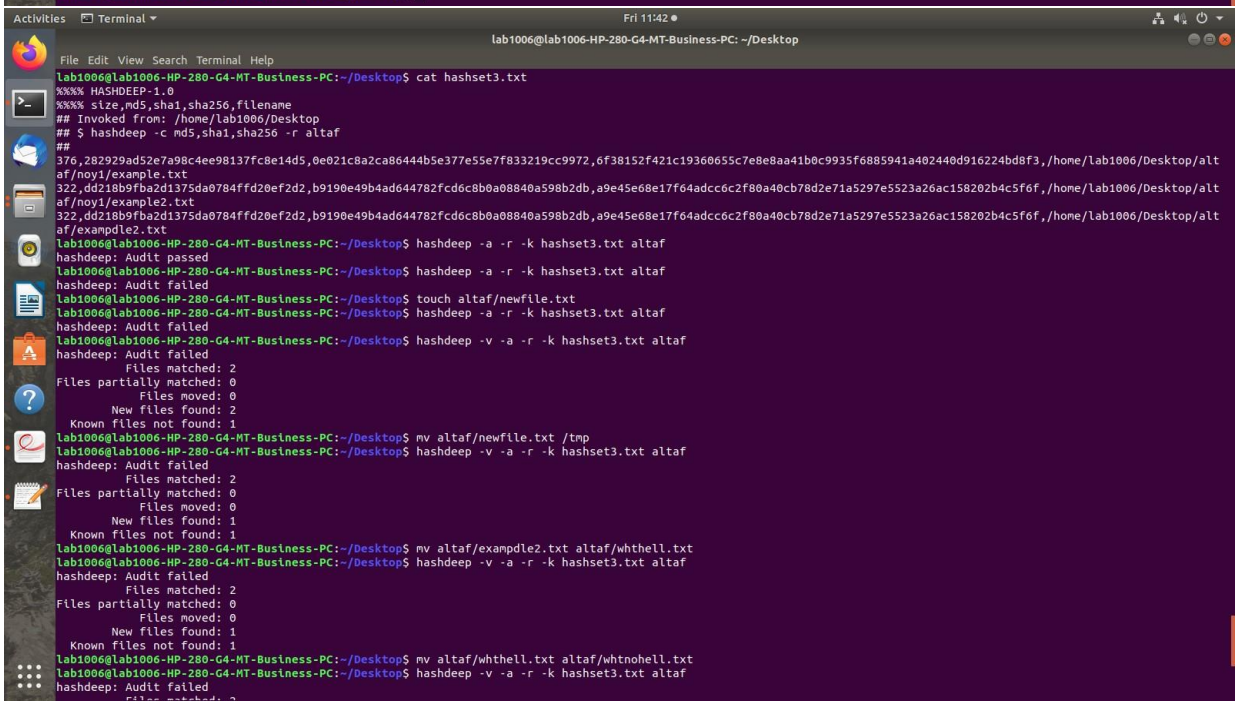
In this example, the generated hash values will be stored in `hash.txt`, and by matching and auditing, you can ensure the integrity of your files.

Output:

Name : Altaf Alam

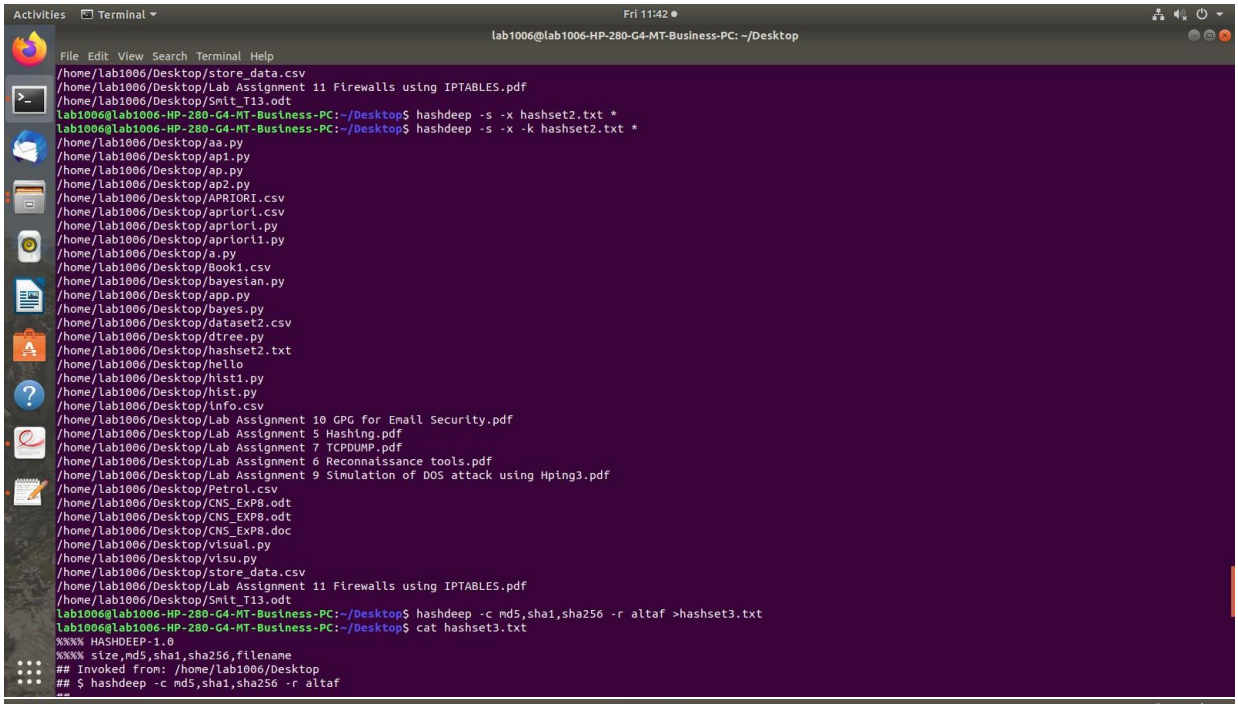
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

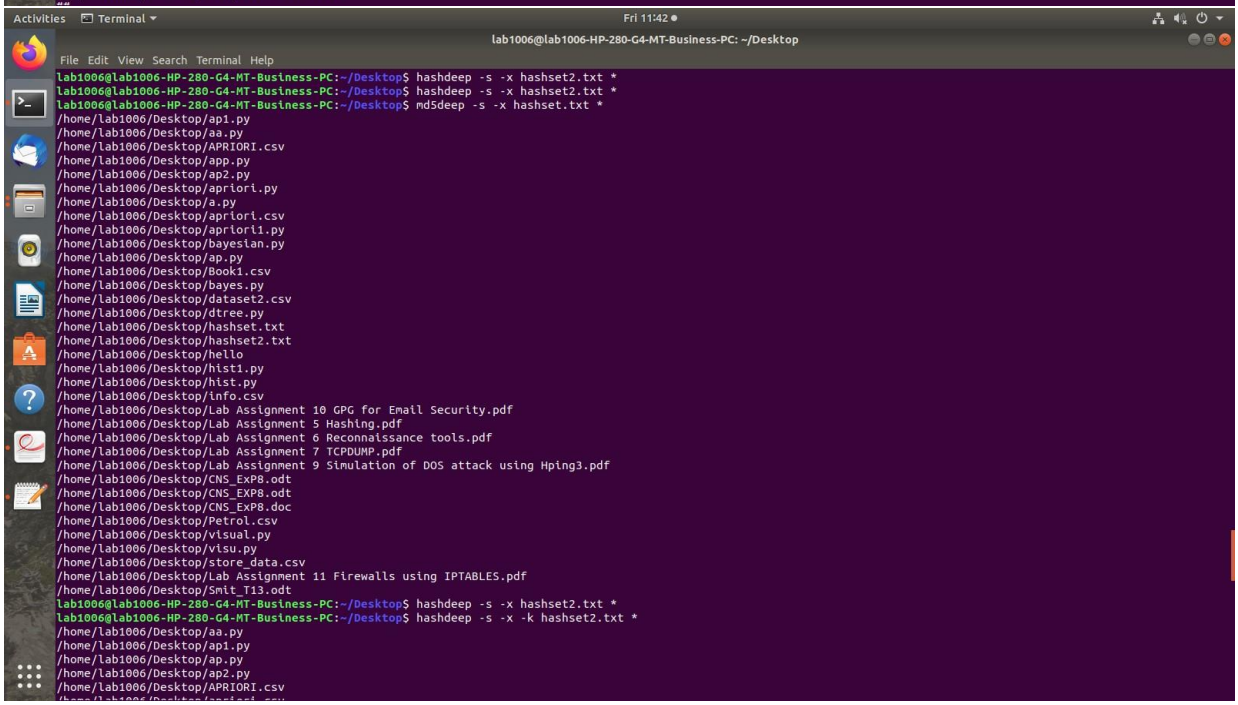




Name : Altaf Alam

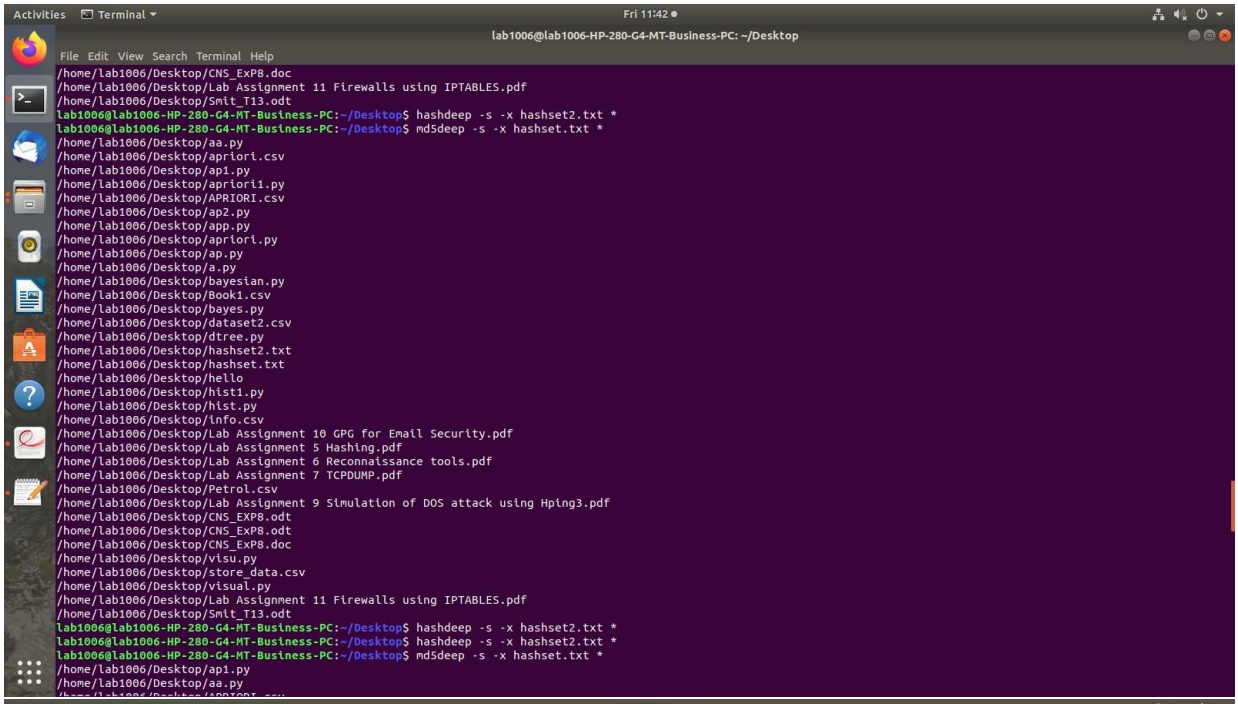
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

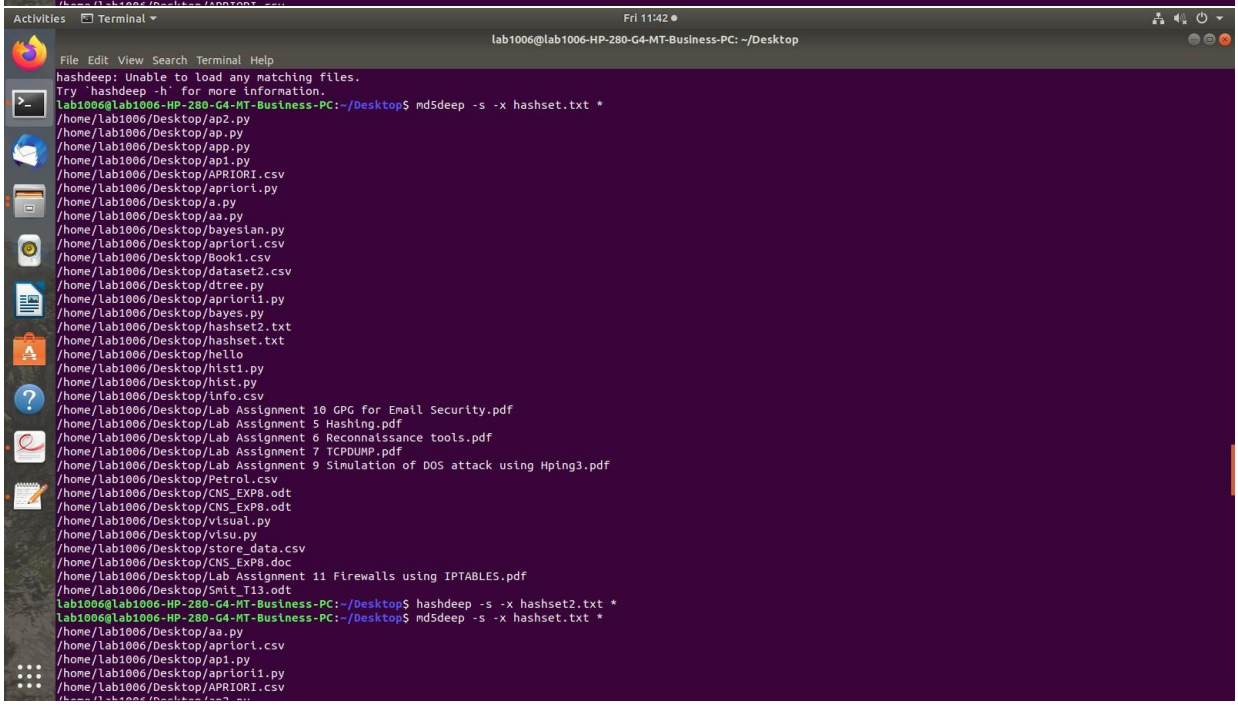




Name : Altaf Alam

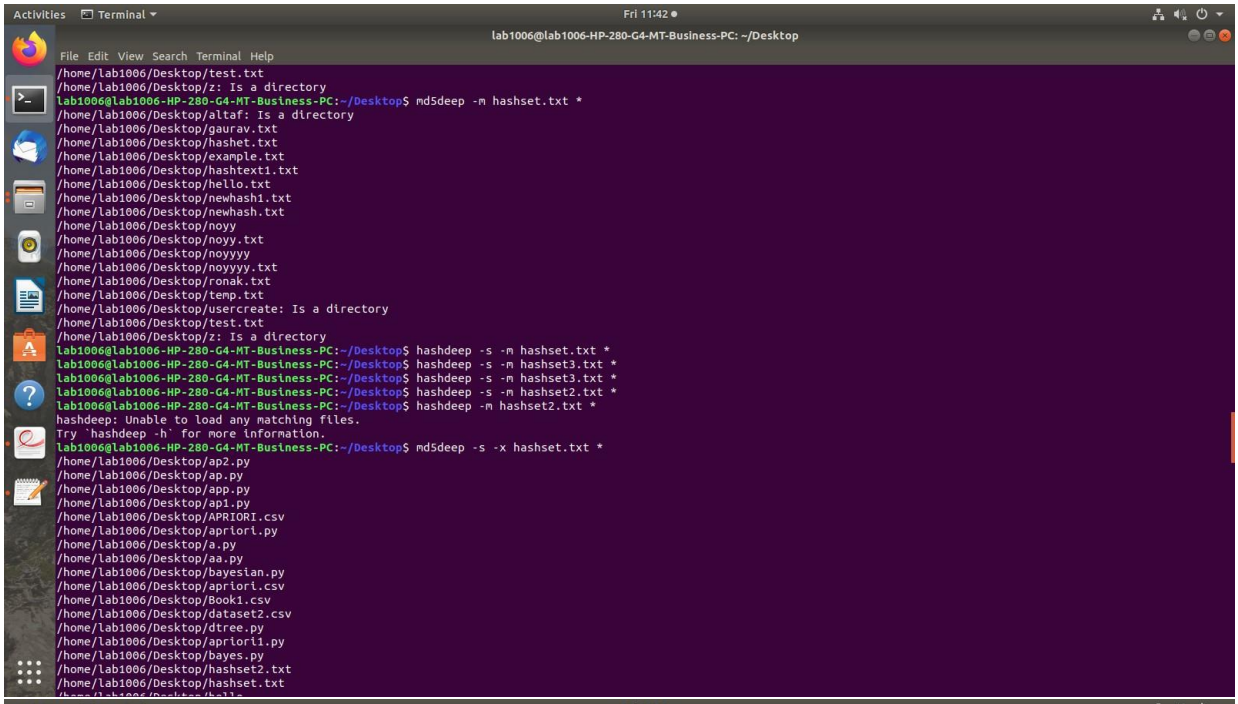
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

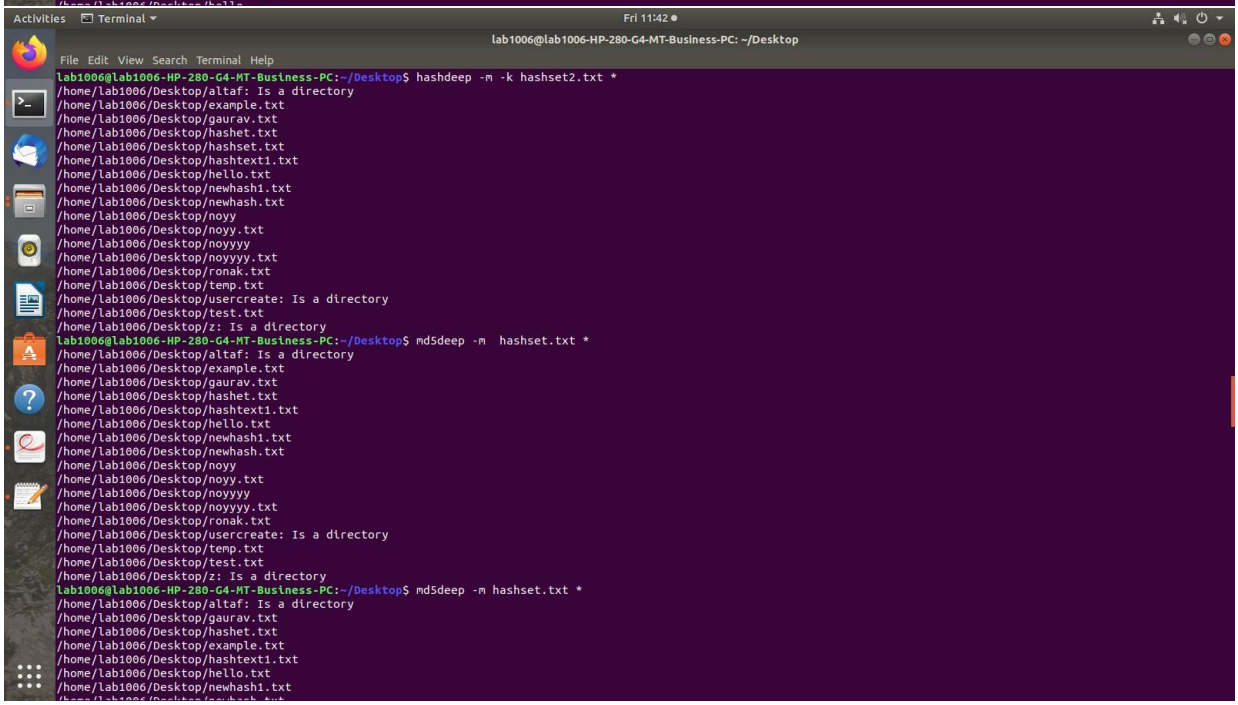




Name : Altaf Alam

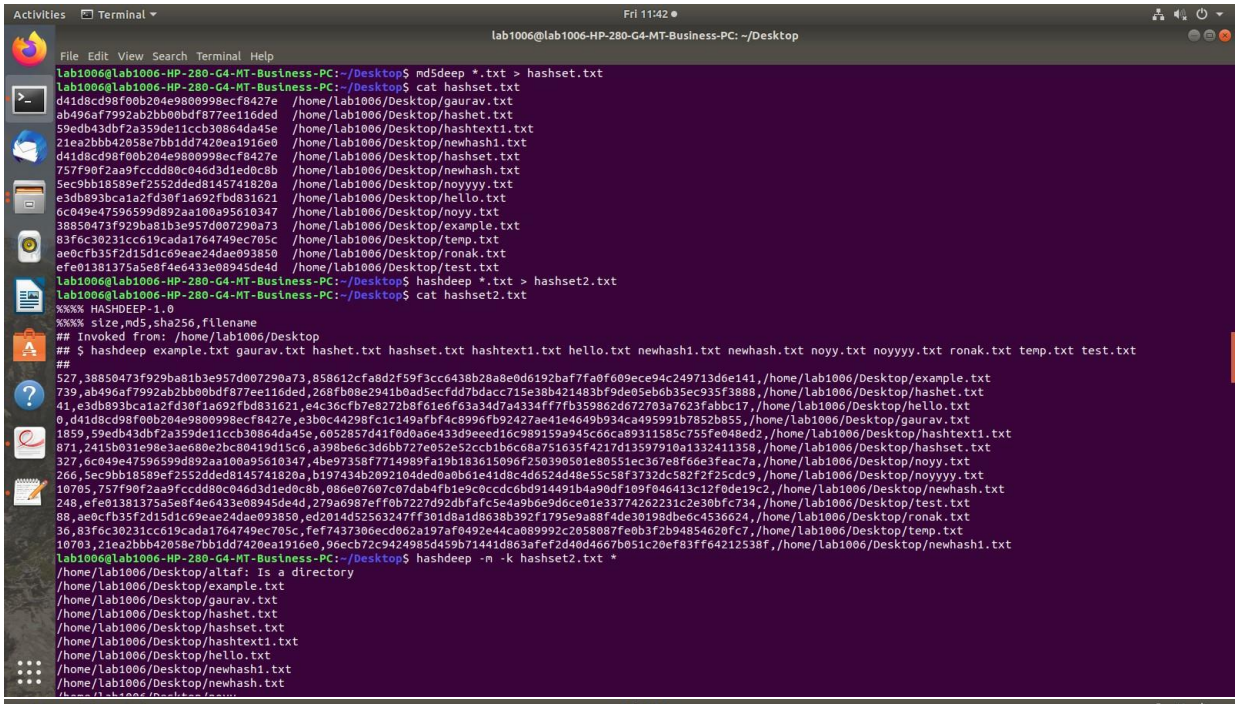
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

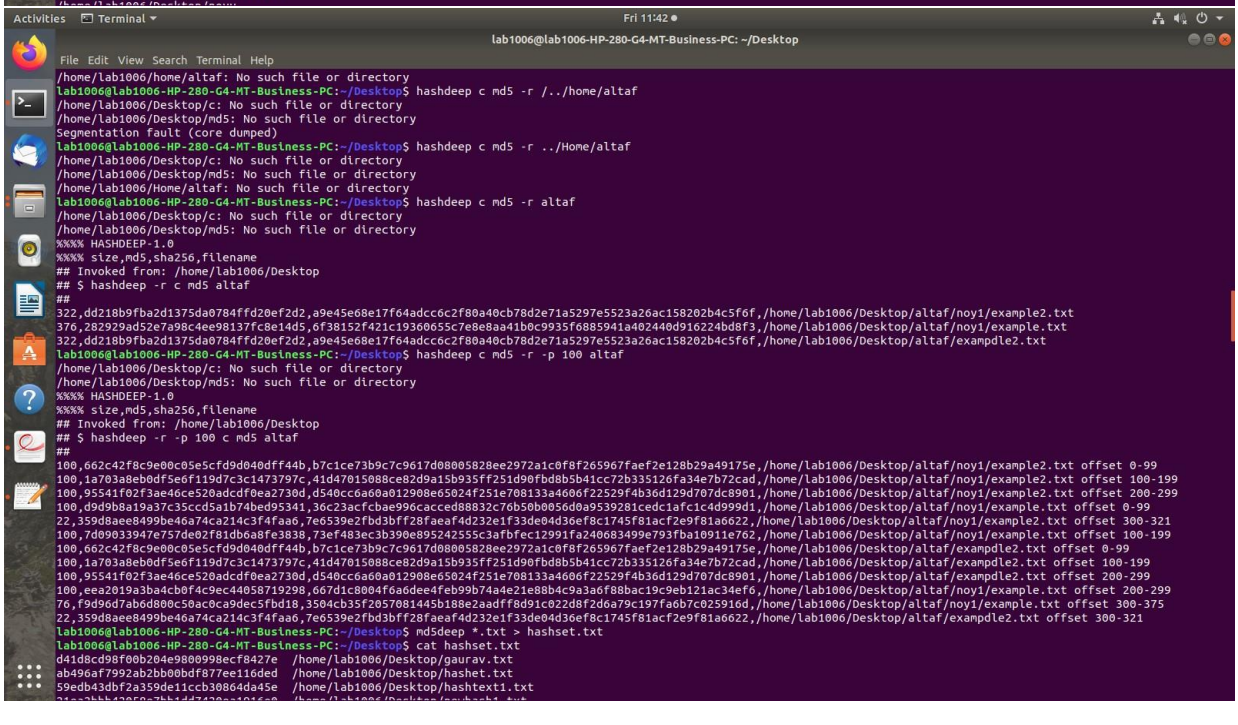




Name : Altaf Alam

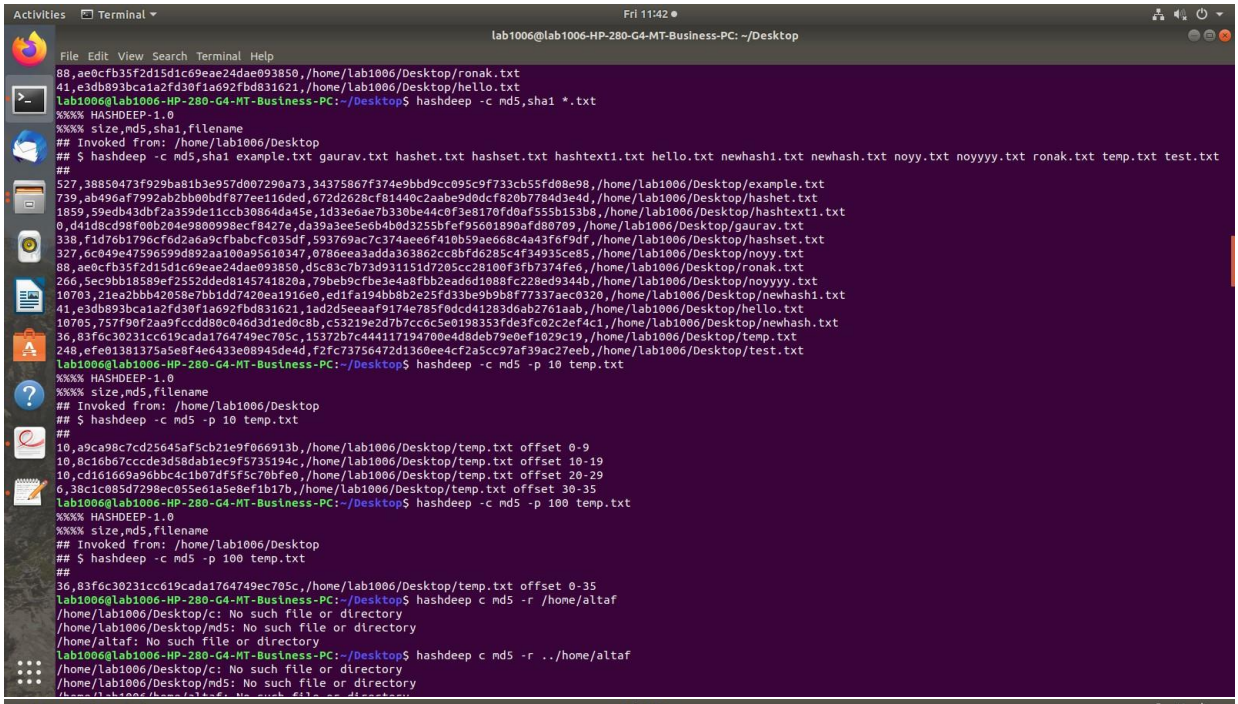
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

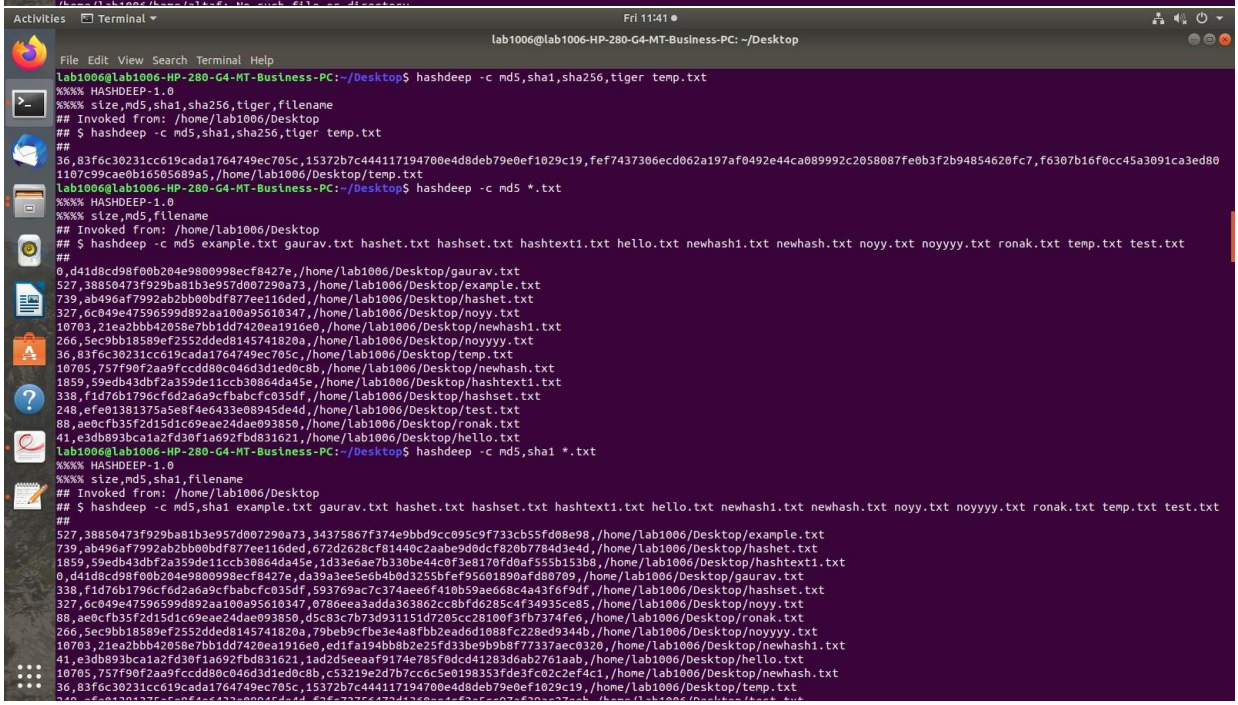




Name : Altaf Alam

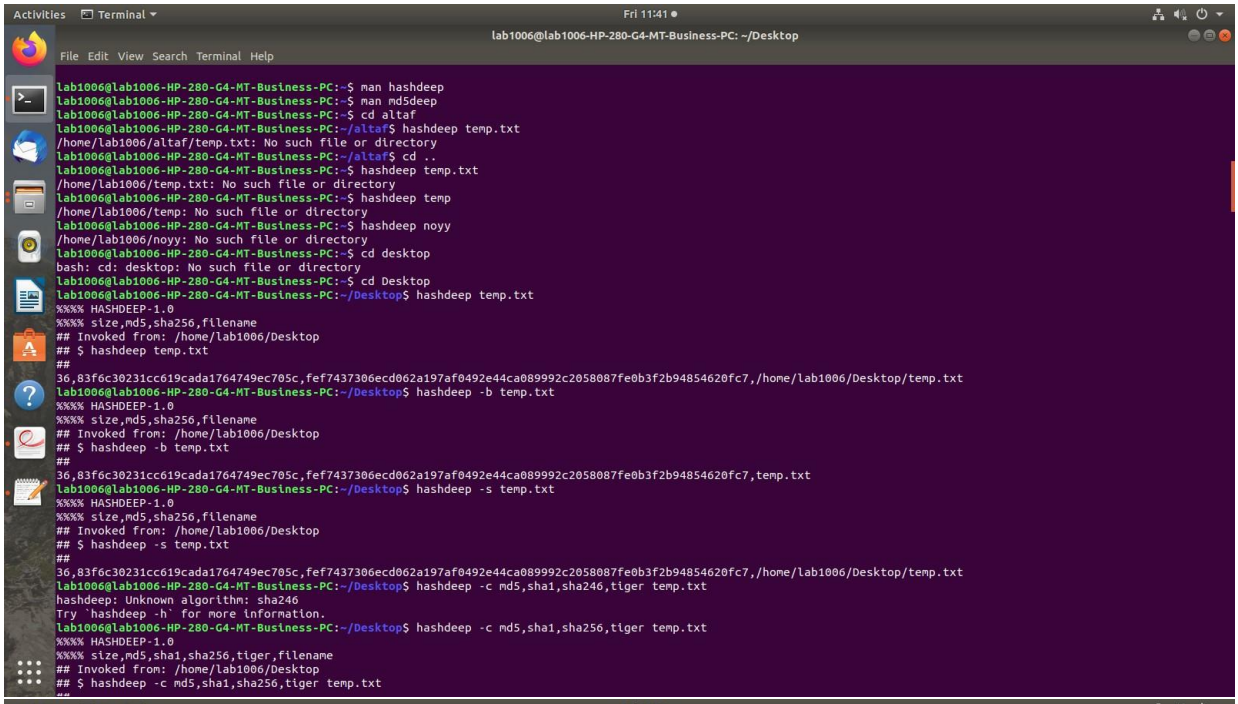
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23

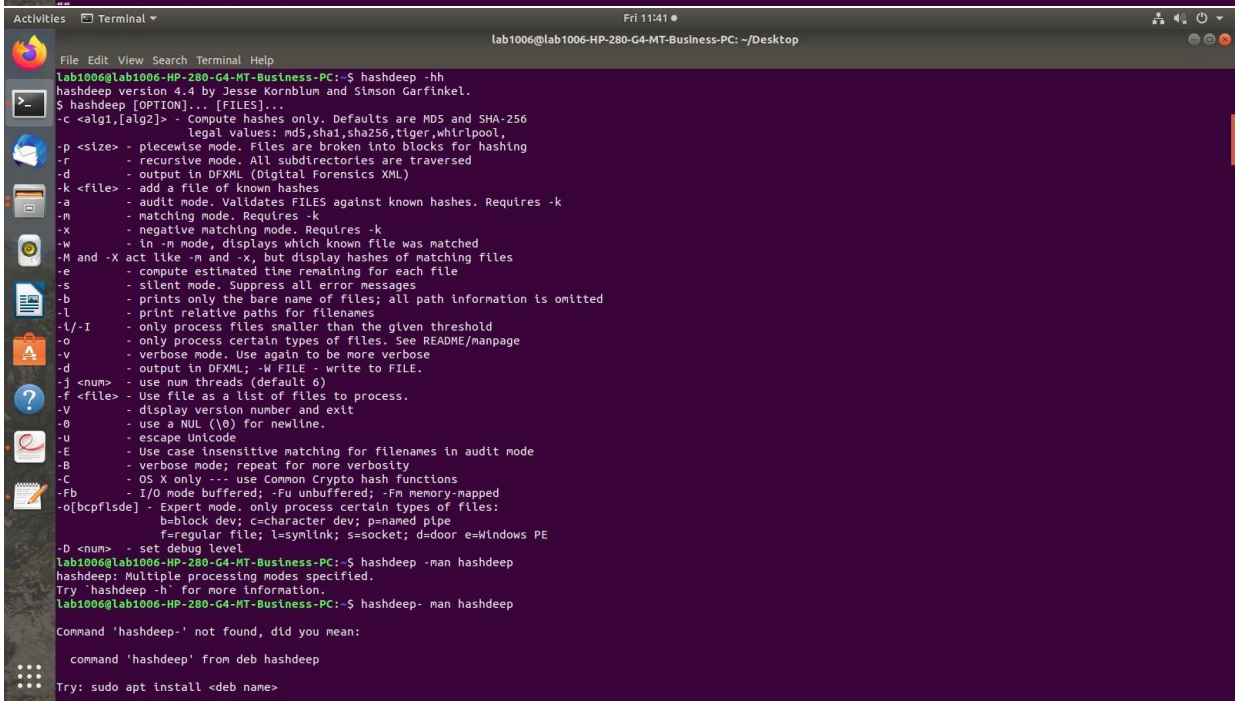




Name : Altaf Alam

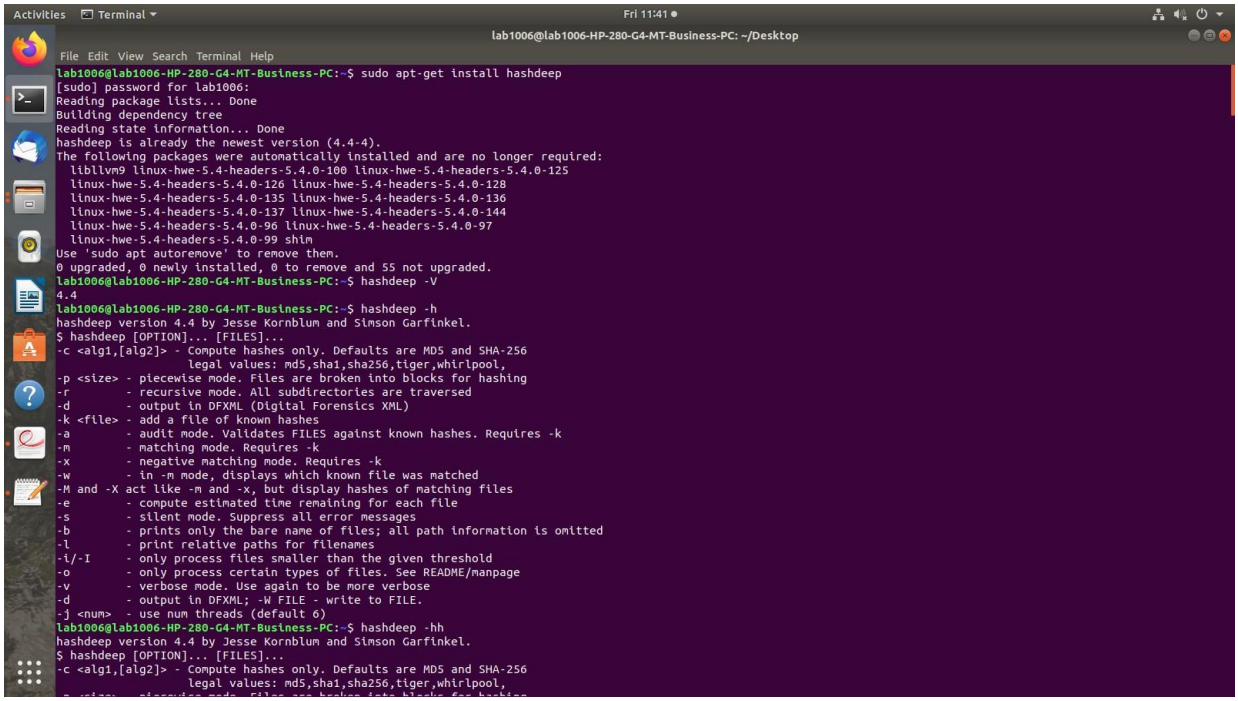
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23





Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 19/08/23



Conclusion:

In conclusion, hashing is essential for various computer science applications, providing efficient data retrieval, security, and integrity. Different hashing algorithms cater to different requirements, ranging from fast data distribution to cryptographic security. The choice of hashing algorithm depends on the specific use case, balancing factors like speed, security, and suitability for the task at hand.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

Experiment 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 :

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

Theory :

Q1)What is the important information that attackers look for using whois command and what attacks can be performed using this information?

WHOIS Command: Unmasking Critical Information and Vulnerabilities

The WHOIS command is a valuable tool that allows users to query a database to retrieve essential information about domain names and IP addresses. While it serves legitimate purposes, attackers can exploit the information gained from WHOIS queries to plan and execute various cyber attacks. Understanding the crucial data attackers seek and the potential attacks they can launch is vital for enhancing cybersecurity.

Important Information Attackers Seek:

1. Domain Ownership Details: Attackers look for the contact details of domain owners, including names, addresses, email addresses, and phone numbers. This information can be exploited for spear-phishing and social engineering attacks.

2. Domain Expiry and Registration Dates: Knowing the registration and expiry dates helps attackers determine how long a domain has been active. New domains might indicate potential malicious activity, while an expiring domain could be a target for domain hijacking.

3. Administrative and Technical Contacts: Attackers can target less secure administrative or technical contacts to gain unauthorized access to the domain's control panel or hosting account.

4. Name Server Information: The name server details can reveal the hosting provider and domain's infrastructure, providing attackers with insights into the underlying technology.

5. DNS Records: Attackers can identify subdomains and other records (like MX records) that could be exploited for phishing or targeted attacks.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

Attacks Using WHOIS Information:

1. Spear-Phishing and Social Engineering: Armed with the domain owner's email address and contact information, attackers can craft personalized spear-phishing emails that appear legitimate. They can trick victims into revealing sensitive information or clicking on malicious links.

2. Domain Hijacking: If attackers identify an expiring domain, they might try to register it as soon as it becomes available. Once they control the domain, they can redirect traffic to malicious websites or launch phishing attacks.

3. Identity Theft: Attackers can use publicly available contact details for domain owners to create convincing impersonation schemes, causing reputational damage or financial loss.

4. Targeted Attacks: Knowing the technical details, hosting provider, and other infrastructure information allows attackers to tailor attacks specifically to the target's technology stack.

5. Reconnaissance for Exploits: Attackers might exploit known vulnerabilities in the domain's hosting provider or infrastructure based on the gathered technical information.

6. Physical Attacks: In some cases, attackers might exploit personal address information for physical threats, harassment, or extortion.

Defensive Measures:

To mitigate the risks associated with WHOIS information exposure, domain owners and organizations can take several steps:

1. WHOIS Privacy Services: Use WHOIS privacy services to shield sensitive information from public view by replacing personal details with proxy contact information.

2. Regular Monitoring: Continuously monitor domain expiry dates to prevent unauthorized renewal attempts.

3. Secure Contacts: Ensure that administrative and technical contacts are secured with strong passwords and two-factor authentication.

4. Security Awareness: Train employees and individuals about the risks associated with sharing personal information online and how to identify phishing attempts.

In conclusion, while the WHOIS command offers essential domain-related information, attackers exploit it to gather critical details for various cyber attacks. Raising awareness about the risks associated with WHOIS information and implementing protective measures are crucial steps in defending against these threats.

Q2)How traceroute command works in order to trace the route of given host?

The `traceroute` command is a network diagnostic tool used to trace the route that packets take from your computer to a destination host on a network, such as a website or server. It helps you identify the path that network traffic follows and the IP addresses of the intermediate routers or switches it passes through. This information can be crucial for troubleshooting network connectivity issues or understanding the network topology. Here's how the `traceroute` command works:

1. Sending ICMP or UDP Packets: The `traceroute` command works by sending a series of packets to the destination host. It uses either ICMP (Internet Control Message Protocol) or UDP (User Datagram Protocol) packets with gradually increasing Time to Live (TTL) values. The TTL value represents the maximum number of hops (routers or switches) a packet can pass through before being discarded. Each intermediate device decrements the TTL value by 1 before forwarding the packet.

2. Recording Responses: As the packets travel through the network, each intermediate device decrements the TTL value. If the TTL reaches zero, the device discards the packet and sends an ICMP "Time Exceeded" message back to the source (the `traceroute` tool). The source can then determine the IP address of the device that discarded the packet.

3. Hop-by-Hop Discovery: By sending multiple packets with increasing TTL values, `traceroute` can discover the sequence of devices the packets pass through. The first packet will likely reach the first router, the second packet might reach the second router, and so on. This process continues until the packets reach the destination host.

4. Displaying Results: The `traceroute` command displays the results of each packet's journey in terms of IP addresses, hostnames (if available), and round-trip times. The round-trip time is the time it takes for a packet to travel from your computer to an intermediate device and back.

This information helps you analyze the latency introduced by each hop.

5. Completion: Once a packet successfully reaches the destination host, the `traceroute` command stops sending packets and displays the complete route from your computer to the destination. It typically displays the IP addresses or hostnames of all the intermediate devices the packets traversed. It's important to note that some network devices or firewalls might be configured to not respond to ICMP or UDP packets, which can result in incomplete or

inaccurate `traceroute` results. In such cases, you might see asterisks (*) or timeouts for certain hops.

In summary, the `traceroute` command is a valuable tool for diagnosing network issues and understanding the network path packets take to reach a destination. It provides insights into the topology of the network and helps identify potential bottlenecks or points of failure.

Q3)Explain dig command with various options.

The "dig" (Domain Information Groper) command is a powerful DNS (Domain Name System) tool used to query DNS servers for various types of DNS information. Here's an explanation of some common "dig" command options:

1) Basic Query: dig example.com
This basic form of the command queries the A record (IPv4 address) for the domain "example.com."

2) Querying specific record types: dig example.com MX
This command queries the Mail Exchanger (MX) records for "example.com."

3) Querying a specific DNS server: dig example.com @dns-server
This command queries the DNS server specified by "dns-server" for the A record of "example.com." 4) Querying a specific DNS server with a specific record type:

dig example.com MX @dns-server

This command queries the DNS server for the MX records of "example.com."

5) Reverse DNS lookup (PTR record): dig -x 8.8.8.8
This command performs a reverse DNS lookup for the IP address "8.8.8.8" and returns the corresponding domain name (PTR record). 6) Verbose output: dig +trace example.com

The "+trace" option shows a full trace of the query, displaying the path of authority from the root to the domain "example.com."

7)Querying with a specific DNS port: dig example.com -p 5353
This command queries the domain "example.com" using port 5353 instead of the default port 53.

8)Querying a specific DNS record form a specific nameserver:
dig @dns-server example.com ANY

This command queries the DNS server "dns-server" for all types of DNS records (ANY) for "example.com."

Q4)Explain any two vulnerabilities detected for the website that you have scanned using nikto. Which attacks are possible if these vulnerabilities are exploited? 1) Directory listing: Nikto looks for directory listings that may expose sensitive information.

2) Security headers: Nikto verifies the presence and effectiveness of security-related HTTP headers.

If Nikto detects directory listing vulnerabilities in a web application or server, it means that the server is configured in a way that allows an attacker to view the contents of directories and files that should not be publicly accessible. This can lead to several types of attacks:

1. Sensitive Information Exposure: Directory listing vulnerabilities may allow an attacker to view sensitive files or directories containing sensitive data, configuration files, or database backups. This exposure could lead to unauthorized access to critical information.
2. Path Traversal Attacks: An attacker can use the directory listing information to conduct path traversal attacks. By manipulating the URLs, they may access files or directories outside the intended scope, potentially leading to data disclosure or code execution.
3. Information Gathering: Directory listing provides valuable information to attackers about the server's directory structure, potentially revealing the organization's internal infrastructure, which can aid in further attacks.
4. File Tampering: Attackers can modify files in directories with listing vulnerabilities, potentially injecting malicious code or altering data, leading to various security risks. If Nikto detects security header vulnerabilities in a web application or server, it means that the server is not configured to provide adequate security-related HTTP headers.

Exploiting these vulnerabilities can lead to several types of attacks:

1. Cross-Site Scripting (XSS): Lack of proper Content Security Policy (CSP) headers can allow attackers to inject malicious scripts into web pages, leading to XSS attacks and potentially compromising users' browsers.
2. Clickjacking: If the X-Frame-Options header is missing or improperly configured, attackers can use clickjacking techniques to trick users into clicking on invisible or disguised elements on the webpage, potentially performing unintended actions.
3. Information Leakage: Missing or improperly set HTTP Strict Transport Security (HSTS) headers can expose users to man-in-the-middle attacks, where sensitive data, such as login credentials, can be intercepted and stolen.
4. Session Hijacking: Without proper Secure (HTTP Only) and SameSite cookies, attackers may be able to steal users' session cookies, leading to session hijacking and unauthorized account access.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

Q5)Write commands for email harvesting and subdomain harvesting Email Harvesting Command:

```
theharvester -d example.com -b google
```

This command utilizes "theharvester" tool to perform email harvesting from the target domain "example.com" using Google as the data source. The tool scrapes Google search results to extract email addresses associated with the target domain. Email harvesting can be used for legitimate purposes like gathering contact information or for malicious activities like spamming and phishing.

Subdomain Harvesting Command:

```
theharvester -d example.com -b baidu
```

With this command, the "theharvester" tool is employed again, this time to perform subdomain harvesting on the domain "example.com" using Baidu as the data source. The tool queries Baidu's search results to identify subdomains associated with the target domain. Subdomain harvesting aids in identifying potential entry points, infrastructure, or exposed resources on a target's subdomains, useful for both security assessment and malicious reconnaissance.

Q6)What are different functionalities provided by dmitry. Write Dmitry command for whois lookup, retrieve Netcraft info, search for subdomains, search for email addresses, do a TCP port scan, and save the output to example.txt for the domain example.com

Dmitry: A Swiss Army Knife for Gathering Intelligence

Dmitry is a versatile open-source tool designed for gathering intelligence about a target domain. It provides a range of functionalities that help security professionals, network administrators, and researchers explore various aspects of a domain's online presence. From domain information to subdomains and email addresses, Dmitry offers a plethora of capabilities that aid in understanding and assessing an organization's digital footprint.

Different Functionalities of Dmitry:

1. WHOIS Lookup:
   WHOIS lookup retrieves registration information about a domain. It provides details about the domain owner, registrar, creation and expiration dates, and more.

   Dmitry WHOIS Lookup Command:

   ```bash
   dmitry -winse example.com
   ```

2. IP WHOIS Lookup:
IP WHOIS lookup provides information about the IP address, including the organization associated with it, geographical location, and network contact details.

Dmitry IP WHOIS Lookup Command:

```bash
dmitry -wi example.com
```

3. Retrieve Netcraft Info:
Netcraft provides information about a domain's web server, technology stack, and historical data. This can help identify trends and potential security issues.

Dmitry Retrieve Netcraft Info Command:

```bash
dmitry -wnetcraft example.com
```

4. Search for Subdomains:
Subdomains are prefixes to a domain (e.g., blog.example.com). Searching for subdomains helps identify other online resources associated with the target domain.

Dmitry Search for Subdomains Command:

```bash
dmitry -ws example.com
```

5. Search for Email Addresses:
Email addresses associated with the domain can be useful for communication or analysis. This functionality retrieves email addresses from public sources.

Dmitry Search for Email Addresses Command:

```bash
dmitry -we example.com
```

6. TCP Port Scan:
TCP port scanning helps identify open ports on a domain's IP address, revealing potential services or entry points for further analysis.

Dmitry TCP Port Scan Command:

```bash
dmitry -p example.com
```

7. Saving Output to a File:
To save the output of any command to a file, you can use the redirection operator (`>`).

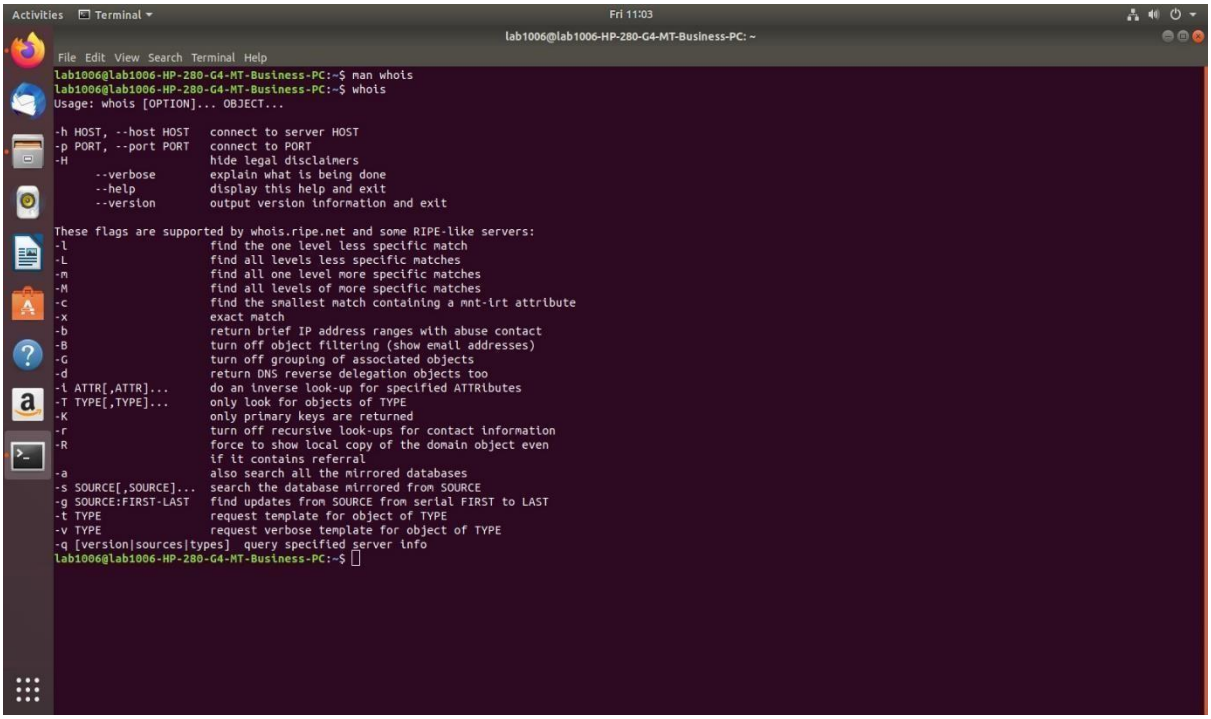
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

Example of Saving Output to a File:

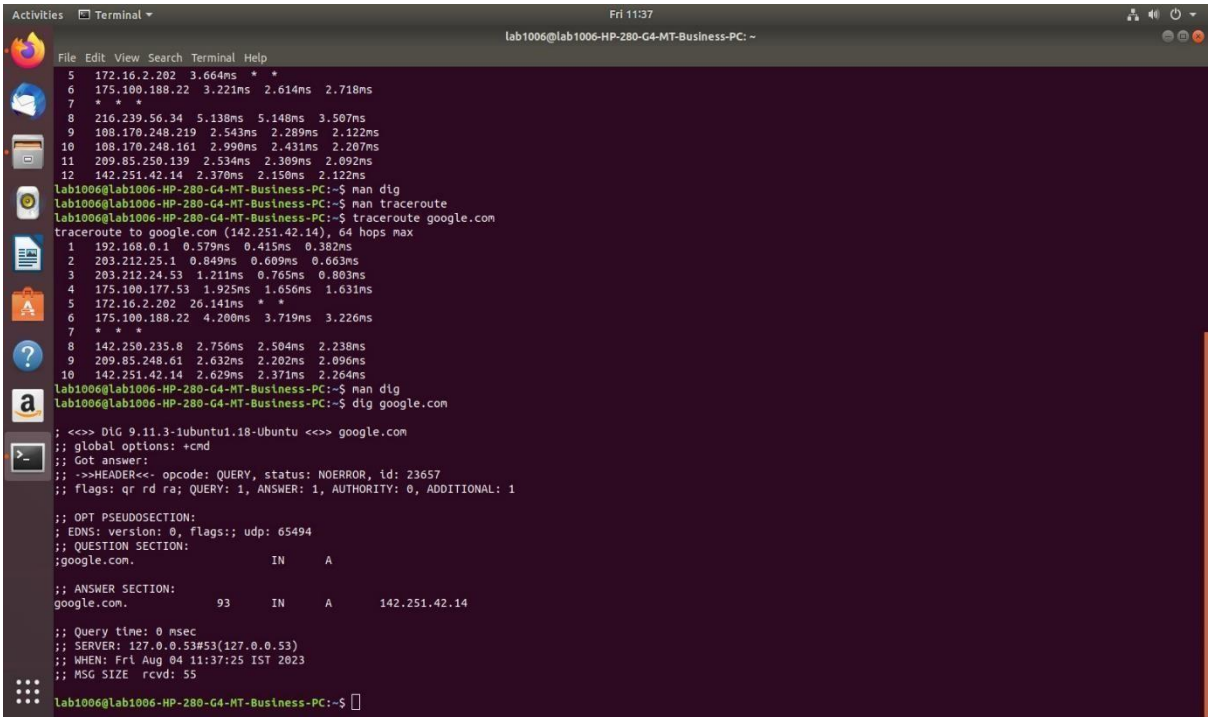
```bash dmitry -winse example.com > example.txt
```

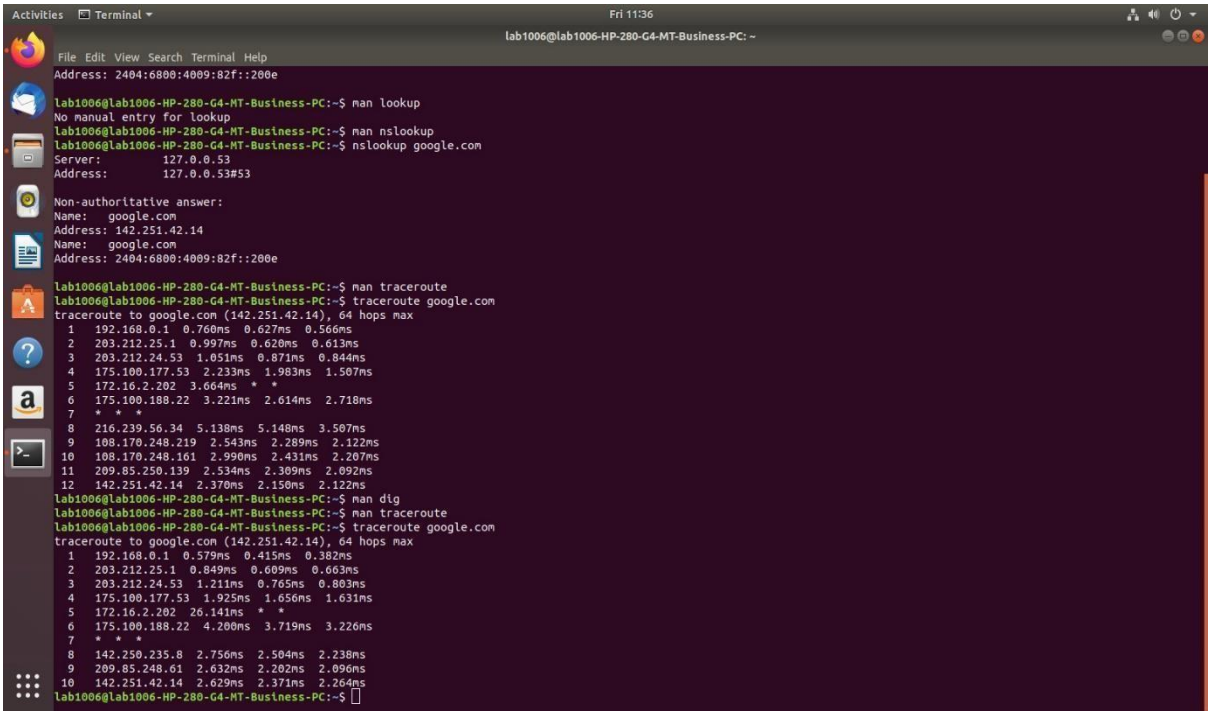
Output:



Name : Altaf Alam

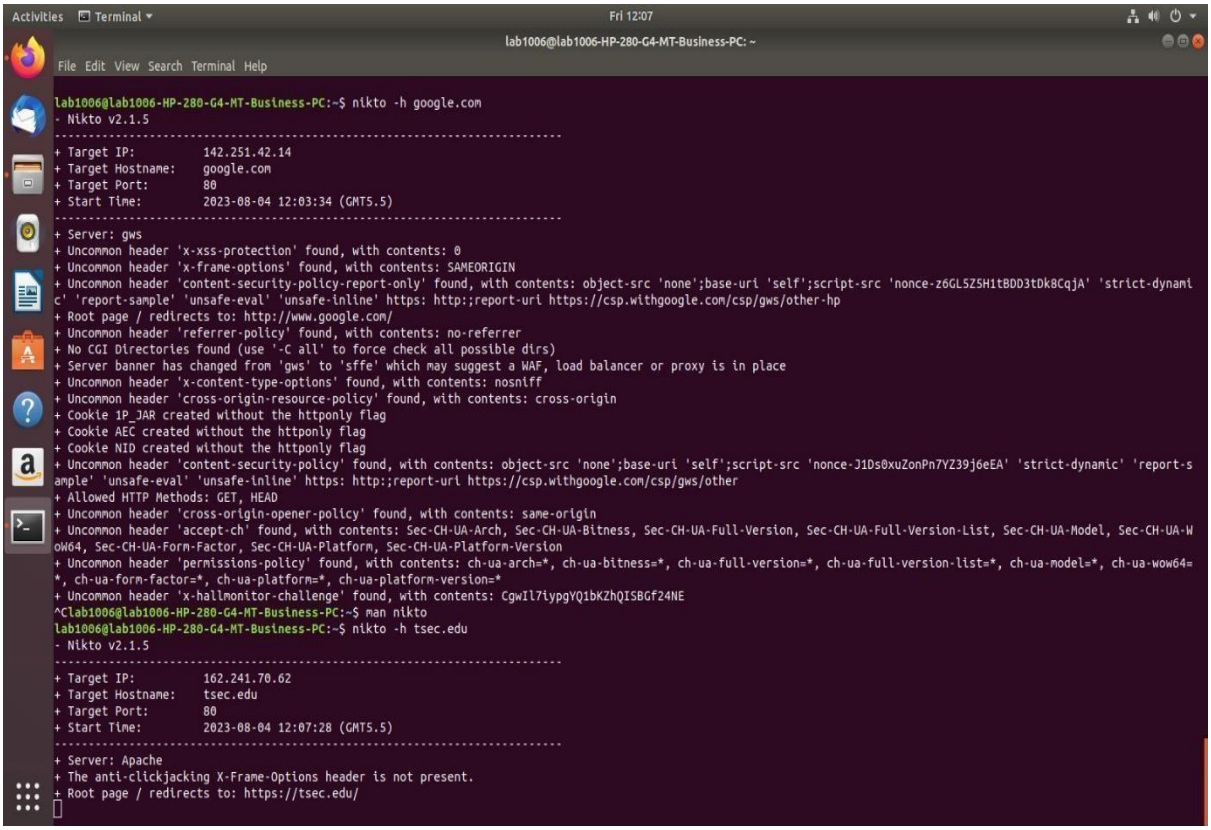
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

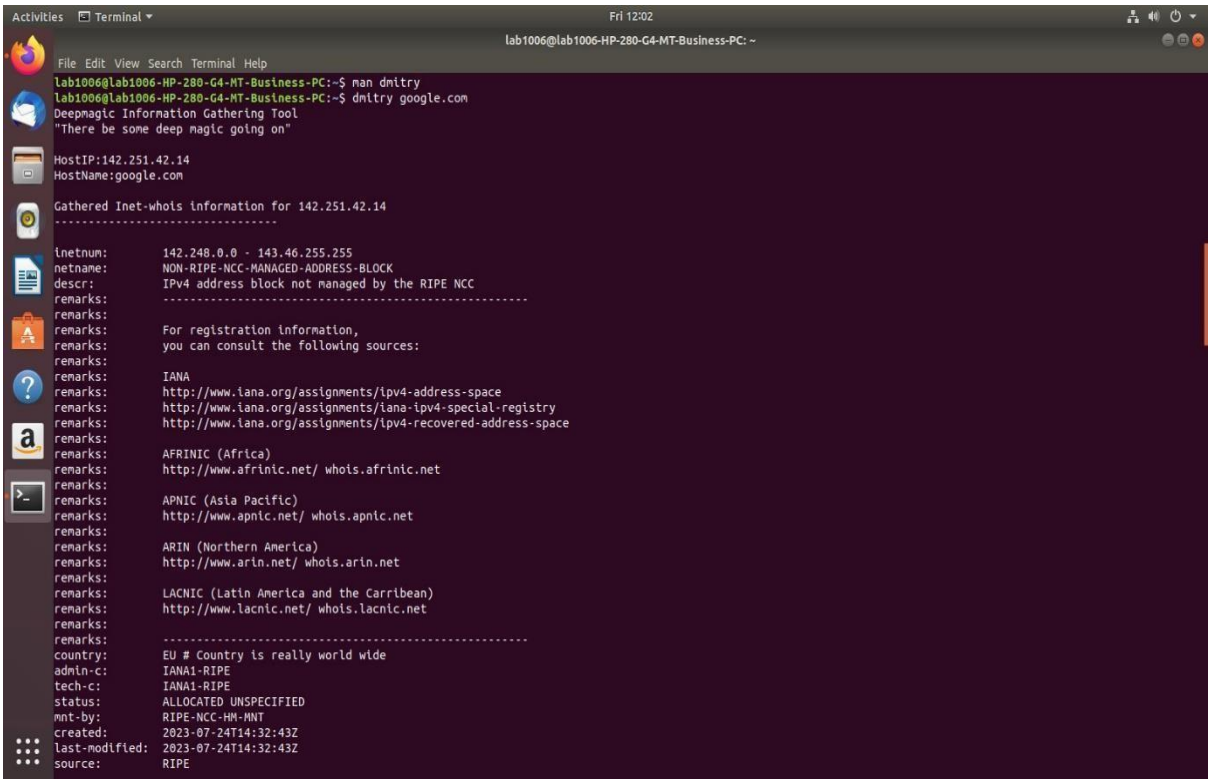




Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23





Conclusion:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 09/08/23

In summary, our exploration of network and web security tools has highlighted their critical roles. Traceroute, Dig, Nikto, and Dmitry offer insights into network paths, DNS information and vulnerability assessment. Ethical and responsible use is essential to leverage these tools for safeguarding systems and data against potential threats.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

Experiment No 7

Aim : Study of packet sniffer tools Tcpdump .

Lab Outcome :

LO3

Theory :

1. What is TCPDUMP and how to install it?

TCPDUMP is a command-line packet capture tool used in Unix-like operating systems to analyze network traffic. It allows users to capture and display the packets that are transmitted and received over a network interface, helping in network troubleshooting, security analysis, and network protocol research.

To install TCPDUMP, follow these steps:

1. Check System Compatibility: TCPDUMP is primarily available for Unix-like systems, including Linux and macOS. Ensure you are working on a compatible system.

2. Package Manager Installation (Linux):
- On Debian/Ubuntu: Open a terminal and use the command `sudo apt-get install tcpdump` to install TCPDUMP using the APT package manager.
- On CentOS/RHEL: Open a terminal and use the command `sudo yum install tcpdump` to install TCPDUMP using the YUM package manager.

3. Homebrew Installation (macOS):
- If you're using macOS and have Homebrew installed, open a terminal and use the command `brew install tcpdump` to install TCPDUMP.

4. Manual Compilation (Optional):
- If you prefer compiling from source, you can download the TCPDUMP source code from its official website (http://www.tcpdump.org) and follow the installation instructions provided in the documentation.

5. Run TCPDUMP:
- Once installed, open a terminal window and use the `tcpdump` command followed by various options and filters to capture and analyze network traffic. For example, `sudo tcpdump i eth0` will capture packets on the "eth0" network interface.

It's important to note that using TCPDUMP typically requires administrative privileges, as capturing network traffic requires direct access to network interfaces. Therefore, using `sudo` before the TCPDUMP command is common practice.

Keep in mind that TCPDUMP can generate a large amount of output, so it's often useful to redirect the output to a file or use filters to focus on specific packets of interest. Always use TCPDUMP responsibly and in compliance with legal and ethical considerations, as network traffic may contain sensitive information.

2. Explain varius commands in tcpdump to capture different types of packets.

TCPDUMP is a versatile command-line packet capture tool that allows you to capture and analyze various types of network traffic. It provides a wide range of options and filters to help you capture specific types of packets for different purposes. Here are some common commands and filters you can use with TCPDUMP to capture different types of packets:

1. Capture on a Specific Interface:

You can specify the network interface to capture packets from using the `-i` option. For example, `tcpdump -i eth0` captures packets from the "eth0" interface.

2. Capture a Specific Number of Packets:

Use the `-c` option followed by a number to capture a specific number of packets. For instance, `tcpdump -c 10` captures the first 10 packets and then stops.

3. Capture All Traffic:

Running `tcpdump` without any options captures all network traffic on the selected interface.

4. Capture by Host or IP Address:

To capture packets from a specific host, use the `host` filter followed by the host's name or IP address. For example, `tcpdump host 192.168.1.1` captures packets to and from the specified host.

5. Capture by Port Number:

You can capture packets based on specific source or destination port numbers using the `port` filter. For instance, `tcpdump port 80` captures packets associated with HTTP traffic.

6. Capture by Protocol:

Use the protocol name (e.g., `icmp`, `tcp`, `udp`) as a filter to capture packets of a specific protocol. For instance, `tcpdump icmp` captures ICMP packets.

7. Capture by Network Range:

You can capture packets from a specific network range using the `net` filter. For example, `tcpdump net 192.168.1.0/24` captures packets from the specified network range.

8. Capture by Packet Size:

Use the `less` or `greater` filters to capture packets smaller or larger than a specified size. For instance, `tcpdump less 100` captures packets smaller than 100 bytes.

9. Capture Specific Traffic Patterns:

Advanced filters allow you to capture specific traffic patterns. For example:

- `tcpdump src host 192.168.1.1 and dst port 80` captures packets from host 192.168.1.1 to port 80.
- `tcpdump src net 192.168.1.0/24 and dst net 10.0.0.0/8` captures traffic between two network ranges.

10. Write Captured Data to a File:

Use the `-w` option followed by a filename to save captured packets to a file. For example, `tcpdump -i eth0 -w capture.pcap` writes captured packets to "capture.pcap" file.

11. Read Captured Data from a File:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

To analyze packets from a previously captured file, use the `-r` option followed by the filename. For instance, `tcpdump -r capture.pcap` reads packets from the "capture.pcap" file.

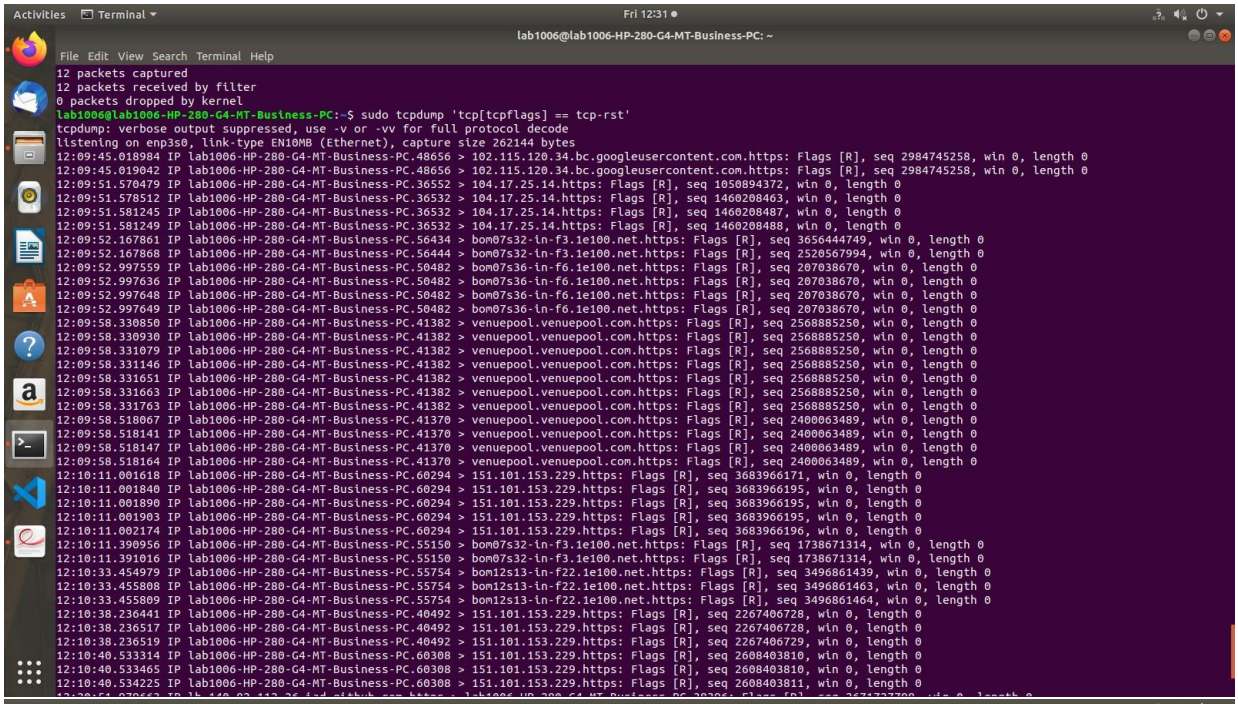
12. Display Captured Data in ASCII:

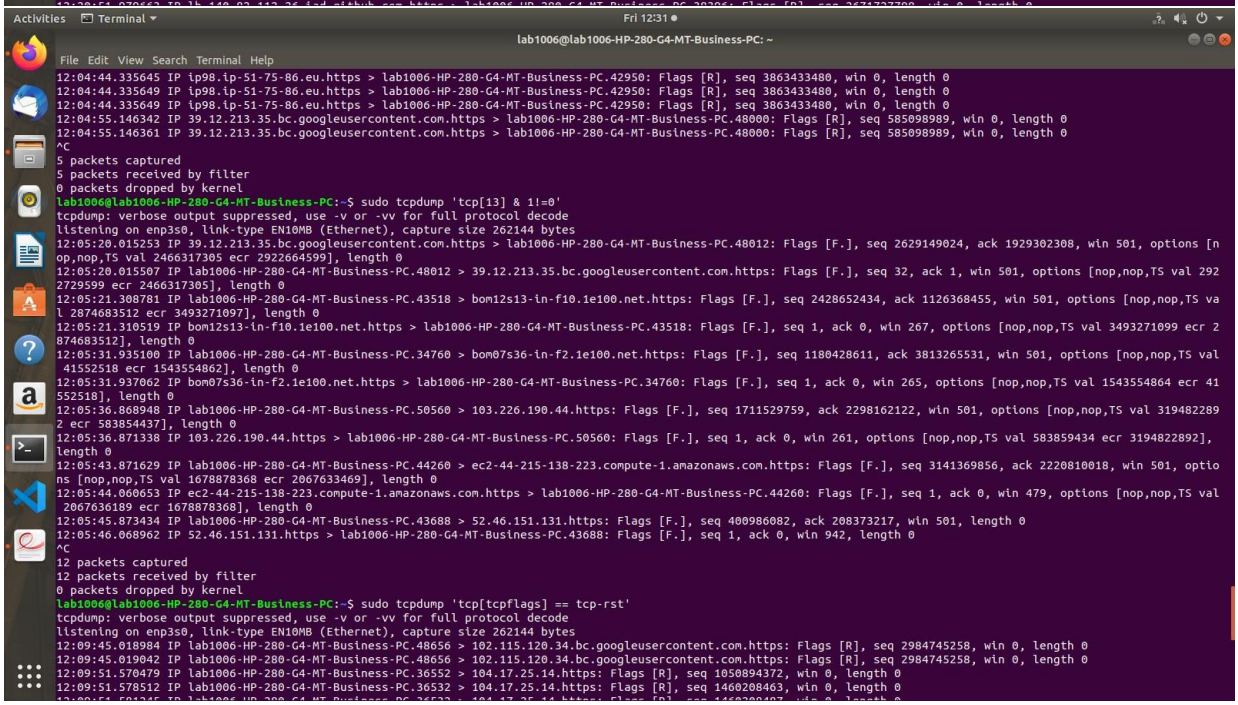
Use the `-A` option to display packet contents in ASCII. This is useful for reading textual data, like HTTP requests and responses.

Output:

Name : Altaf Alam

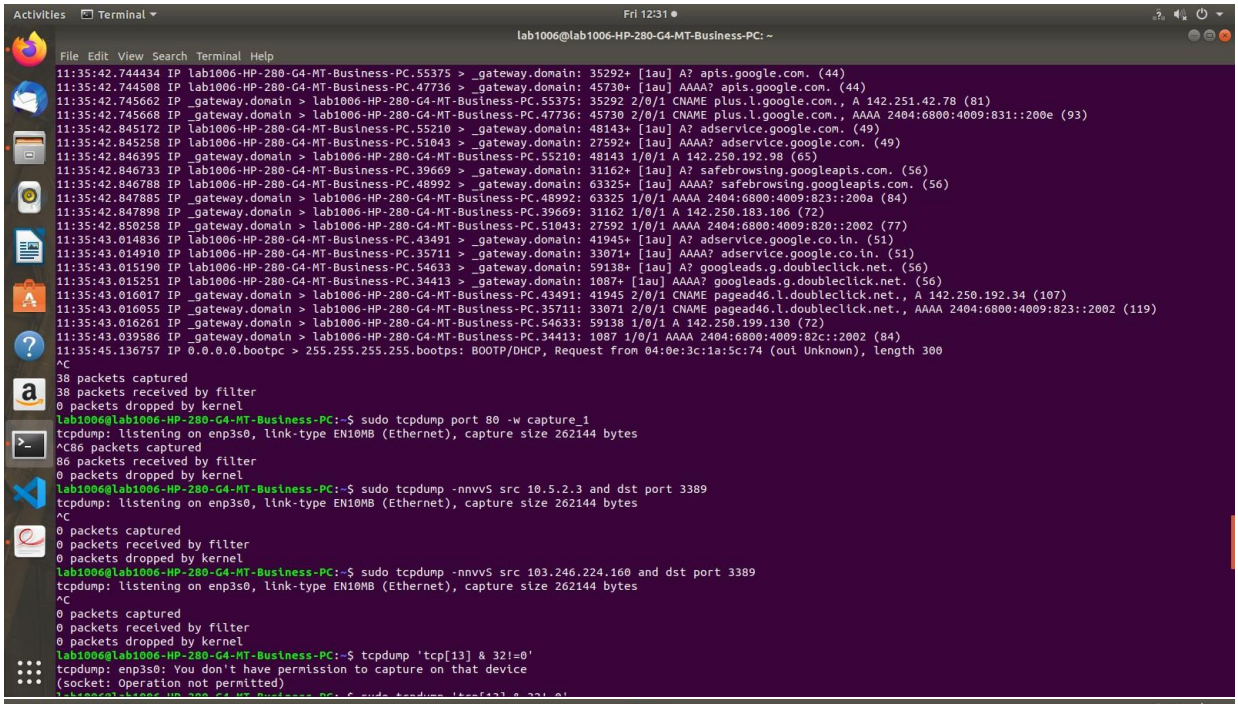
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

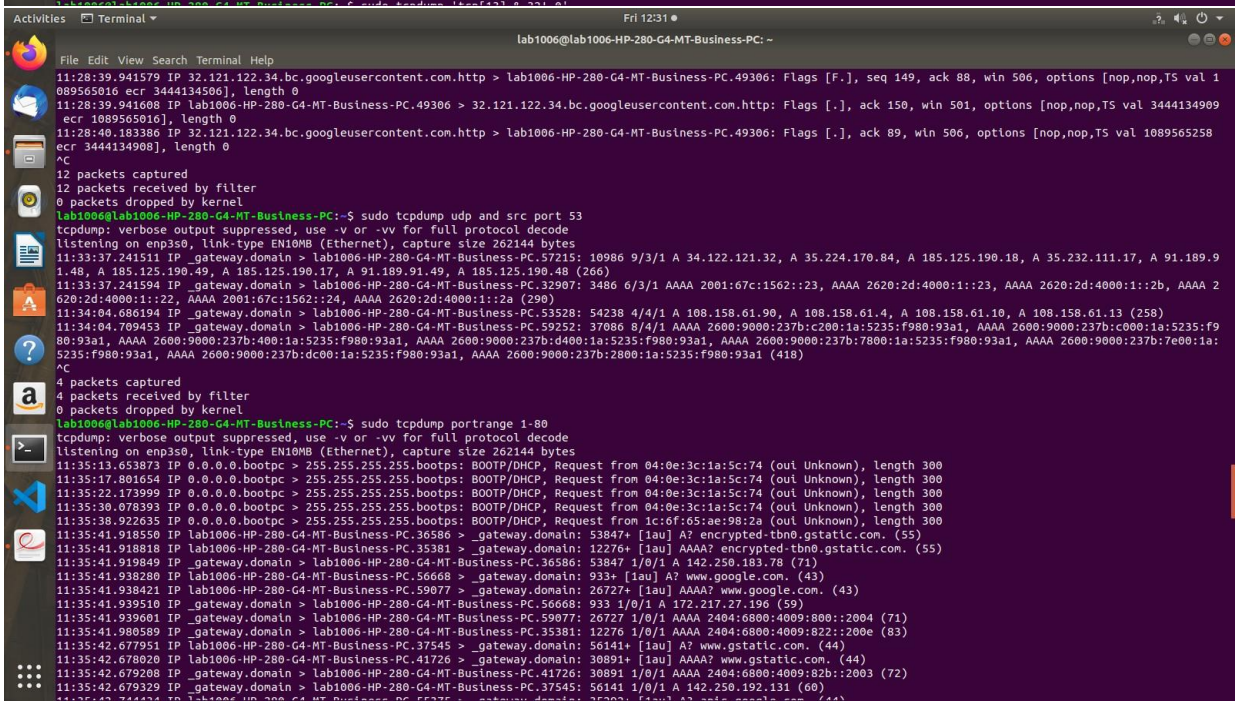




Name : Altaf Alam

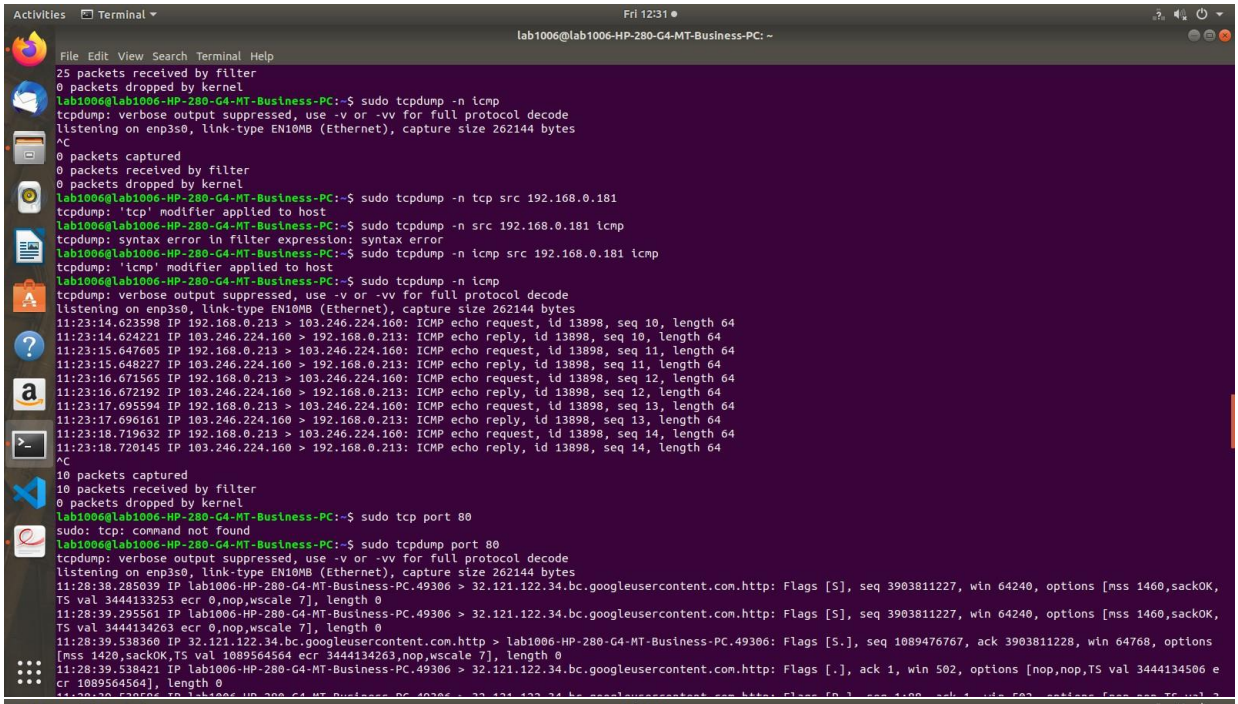
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

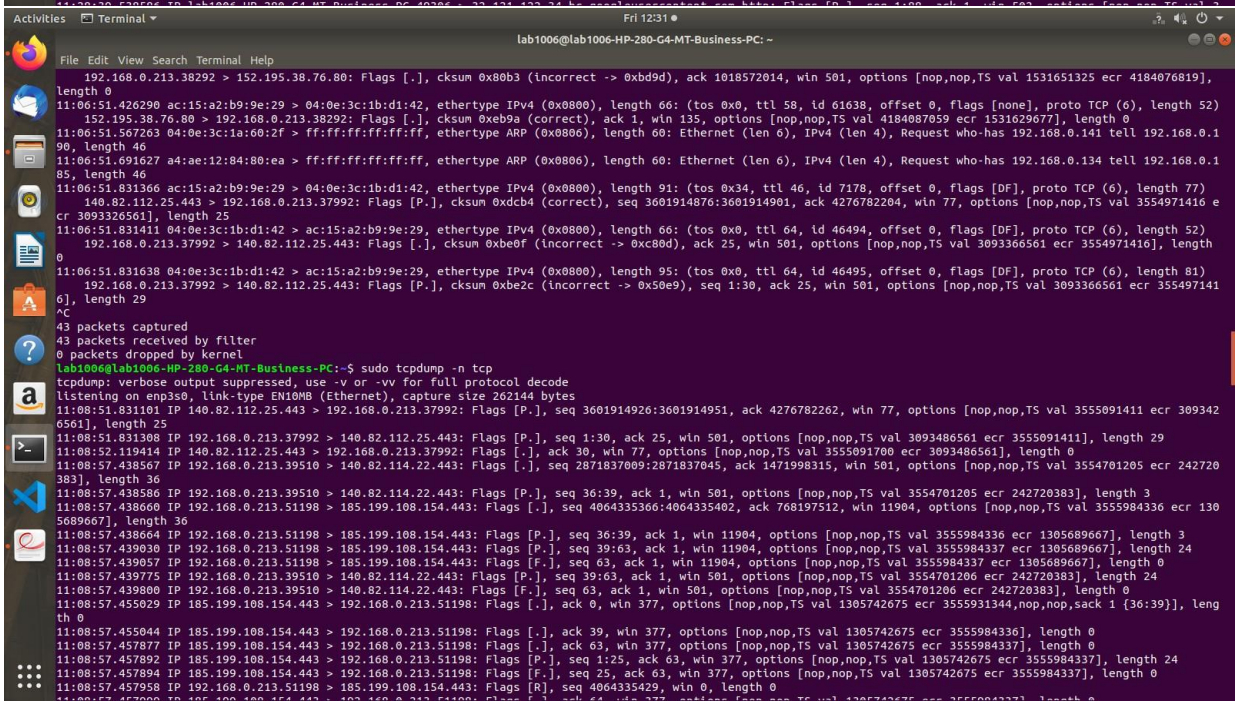




Name : Altaf Alam

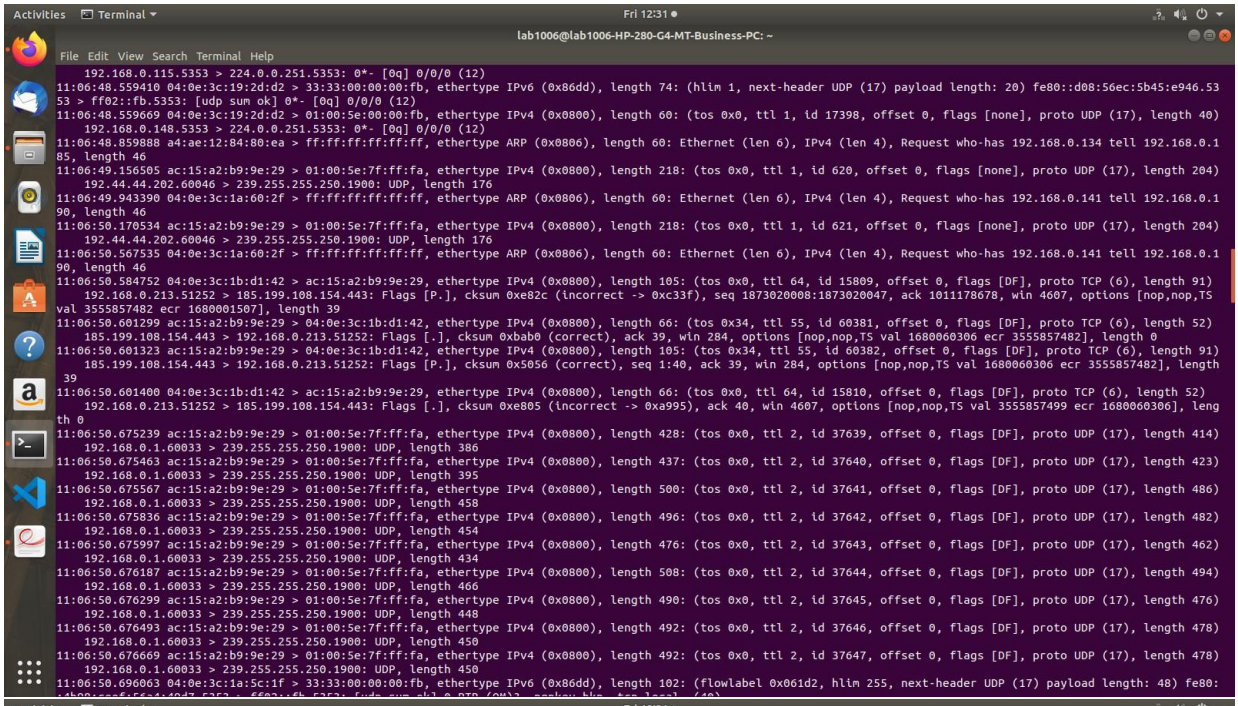
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23

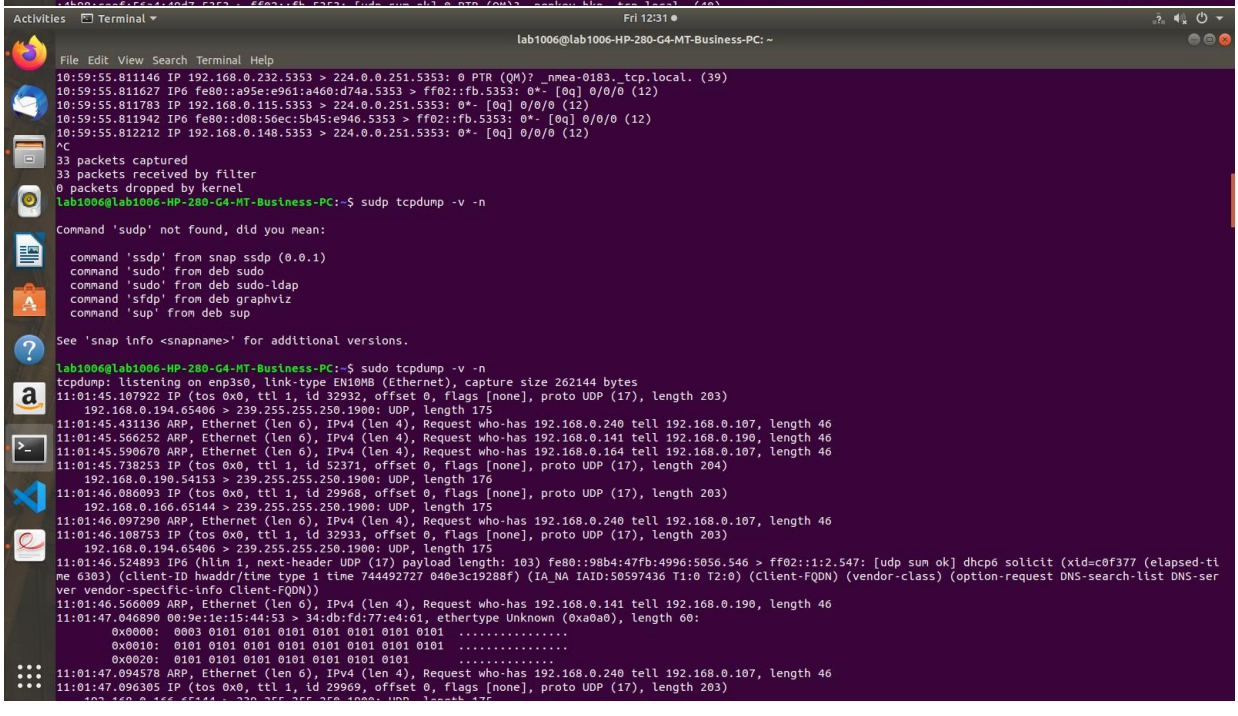




Name : Altaf Alam

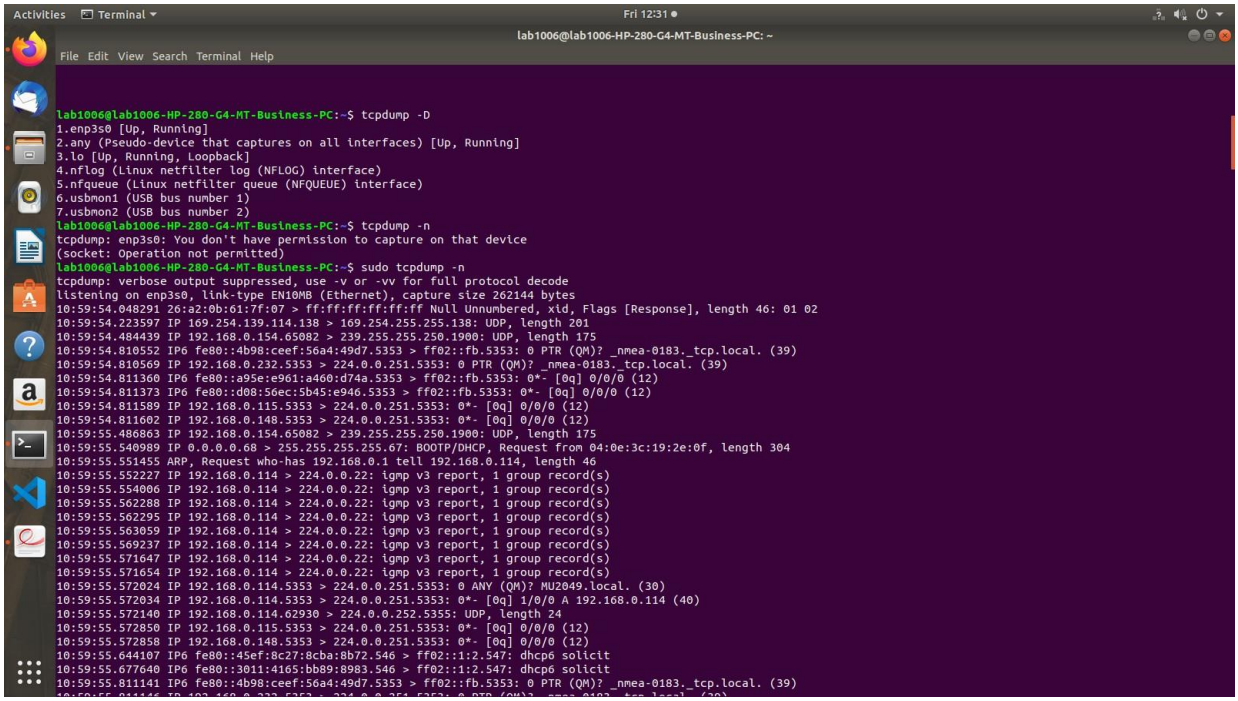
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23





Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 28/08/23



Conclusion:

In conclusion, TCPDUMP is a versatile and powerful command-line tool for capturing and analyzing network traffic. Its various commands and filters enable users to target specific packets based on protocols, hosts, ports, and more. Used responsibly, TCPDUMP is invaluable for network troubleshooting, security analysis, and understanding network communication.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

Experiment 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 :

LO4

Theory :

1. What is port scanning ? What is Nmap?

Port scanning is a crucial technique in the realm of networking and cybersecurity. It involves probing a host or network to discover open ports, which act as gateways for network services or applications. These ports are the entry points through which data flows in and out of a system. Each port is associated with a specific protocol and service, making them essential for communication within a network. The primary purpose of port scanning is to unveil the network landscape, assess the security posture of a system, and identify potential vulnerabilities or attack vectors. Network administrators use it for troubleshooting and monitoring network health, while security professionals use it to detect and mitigate security threats. However, malicious actors also leverage port scanning to identify vulnerable targets for cyberattacks.

Nmap, short for "Network Mapper," is a versatile and widely used open-source tool for network discovery and security auditing. Developed by Gordon Lyon, also known as Fyodor, Nmap has earned a reputation as the go-to tool for port scanning due to its comprehensive feature set and cross-platform compatibility. Nmap's capabilities extend beyond basic port scanning. It can perform a wide range of tasks, including host discovery, service and version detection, operating system fingerprinting, and scripting for automation. Nmap is favored by security professionals, network administrators, and ethical hackers for its flexibility and accuracy.

2. Explain in brief different types of ports?

Ports are essential components of network communication, and they can exist in various states, each conveying specific information about their accessibility and functionality. Understanding these port states is crucial for network administrators and security professionals. Here are brief explanations of different port states:

1. Open: An "open" port is one that is actively listening for incoming connections. It indicates that a network service or application is running and ready to accept data or requests. Open ports are crucial for legitimate network communication.

2. Closed: A "closed" port is one that is not actively listening for connections. It means there is no service or application running on that port. Closed ports are safe from unauthorized access, but they still indicate the presence of a host.

3. Filtered: A "filtered" port is one that cannot be determined as open or closed with certainty. This state often occurs when a firewall, intrusion detection system (IDS), or other security measure blocks incoming requests to the port. It makes it challenging to discern the actual status of the port.

4. Unfiltered: An "unfiltered" port is one that is accessible and can be reached, but its status (open or closed) remains undetermined. Unfiltered ports usually indicate that no significant firewall rules are blocking access to the port.

5. Open | Filtered: This state combines characteristics of both open and filtered ports. It suggests that the port is reachable, but the response to a probing request is filtered, possibly by a firewall. It can be challenging to ascertain the exact state of such ports.

6. Closed | Filtered: This state also combines characteristics of both closed and filtered ports. It implies that the port is accessible, but the response is filtered, typically indicating that a firewall is blocking probing attempts. This state can be confusing during port scanning.

3. Exploring Port Scanning Techniques with Nmap

Port scanning is an integral part of network reconnaissance, allowing us to discover open ports on target hosts and gain insights into their configuration and potential vulnerabilities. Nmap (Network Mapper) is a versatile tool that offers various scanning techniques to achieve this. In this guide, we'll discuss several Nmap scanning techniques and provide commands for each, along with brief explanations of how they work.

1. TCP Connect Scan

Command: `nmap -sT target`

Description: This scan emulates a full TCP connection attempt to each target port. If a connection is successfully established, the port is considered open. It's the most straightforward scanning method, but it can be easily detected by intrusion detection systems (IDS) and firewalls because it fully completes the TCP handshake.

2. TCP SYN Scan

Command: `nmap -sS target`

Description: The TCP SYN scan, also known as the "half-open" scan, is stealthier than the TCP Connect scan. It sends SYN (Synchronize) packets to target ports and examines their responses.

If a port responds with a SYN-ACK (Synchronize-Acknowledgment) packet, it's considered open. If it responds with an RST (Reset) packet, it's considered closed. This scan doesn't complete the full TCP handshake, making it less likely to trigger alarms.

3. FIN Scan

Command: `nmap -sF target`

Description: The FIN scan sends FIN (Finish) packets to target ports. If a port is closed, it should respond with an RST packet. If it's open, it should ignore the FIN packet. This scan is effective for identifying systems with non-standard TCP stack implementations.

4. Null Scan

Command: `nmap -sN target`

Description: Similar to the FIN scan, the Null scan sends packets with no TCP flags set, making them appear "null." If a port is closed, it should respond with an RST packet. If it's open, it should ignore the packet. This technique is stealthy and can bypass some firewall rules.

5. XMAS Scan

Command: `nmap -sX target`

Description: The XMAS scan sets multiple TCP flags in the packet, making it look like a Christmas tree. If a port is closed, it should respond with an RST packet. If it's open, it should ignore the packet. Like the Null scan, this method can bypass firewall rules.

6. ACK Scan

Command: `nmap -sA target`

Description: The ACK scan sends ACK (Acknowledgment) packets to target ports. It can be used to determine if a firewall is in place; open ports will typically respond with an RST packet, while filtered ports may not respond at all. This scan can identify packet-filtering firewalls.

7. Ping Sweep

Command: `nmap -sn target_range`

Description: Ping sweeping is not a port scanning technique but is often used to identify live hosts before conducting port scans. It sends ICMP echo requests (pings) to a range of IP addresses and identifies responsive hosts, reducing unnecessary scanning on non-responsive hosts.

8. Service and Version Detection

Command: `nmap -sV target`

Description: Nmap can probe open ports to identify the services running on them and their versions. This information is crucial for understanding the potential attack surface and vulnerabilities.

9. Port and Port Range Scanning

Command: `nmap -p ports target`

Description: Nmap allows users to specify individual ports or port ranges for scanning, giving flexibility in targeting specific services or performing broad scans of common ports.

10. OS Fingerprinting

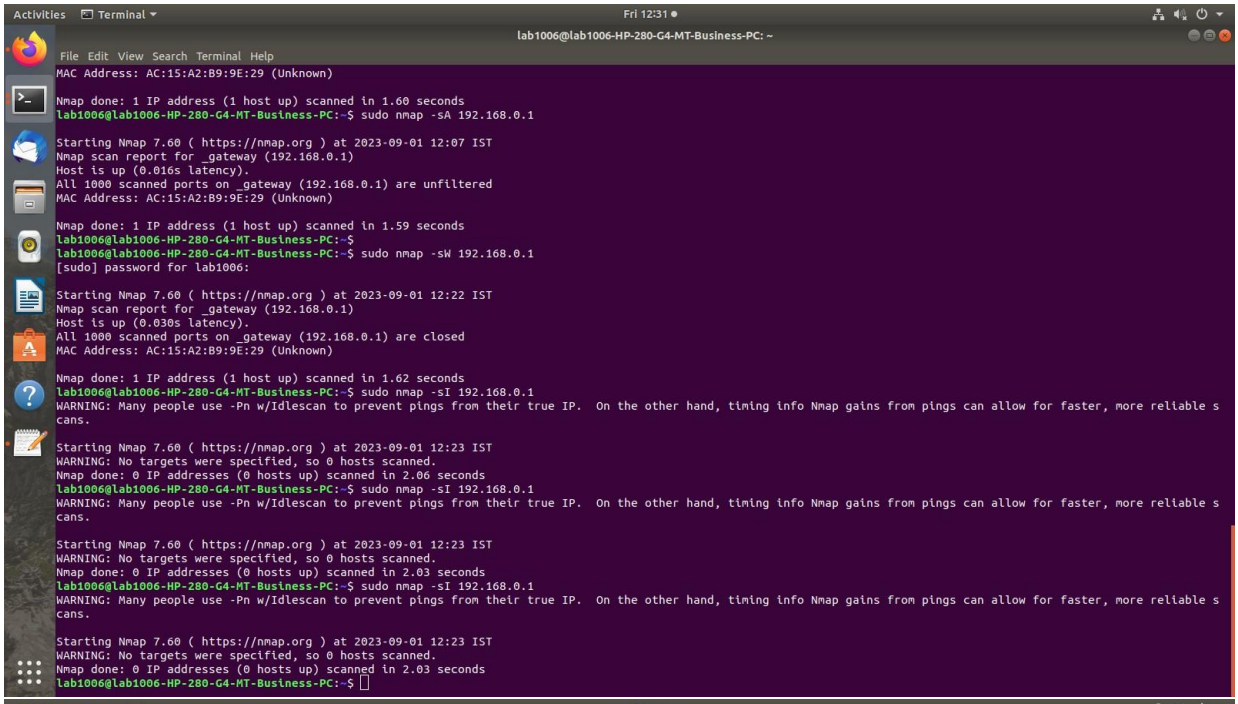
Command: `nmap -O target`

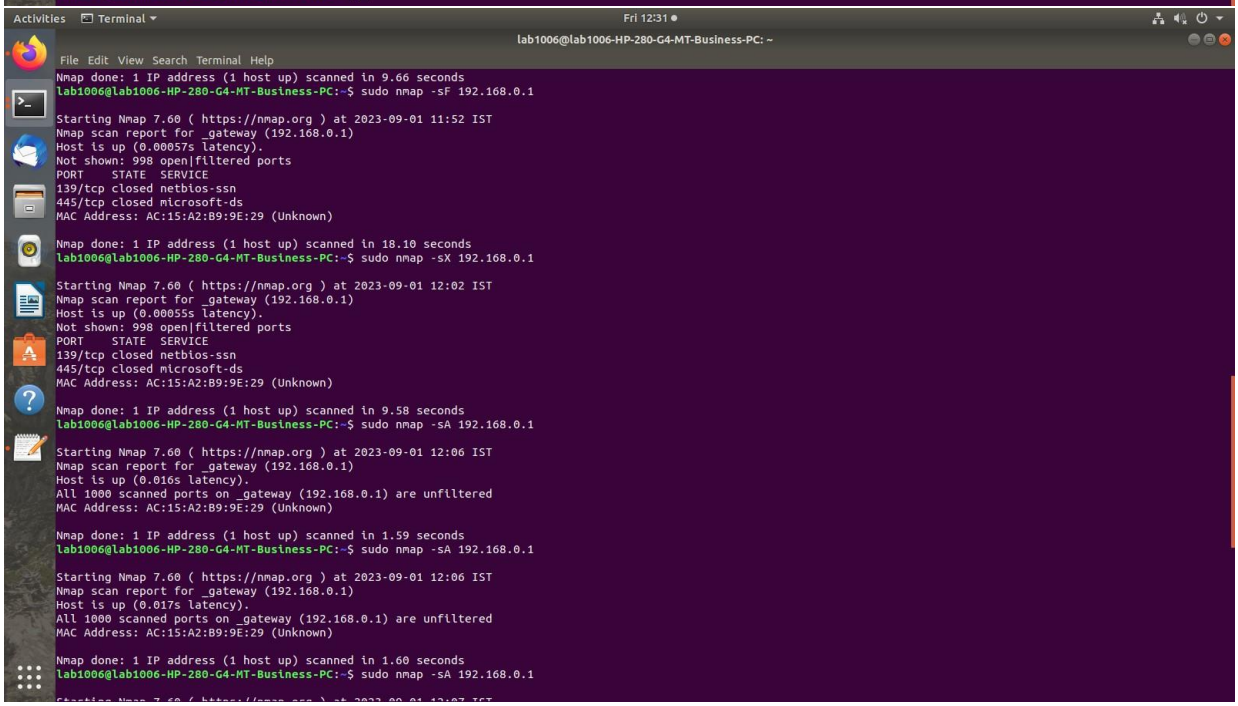
Description: Nmap can attempt to identify the operating system running on the target by analyzing network responses and characteristics. This information aids in understanding the target environment.

Output:

Name : Altaf Alam

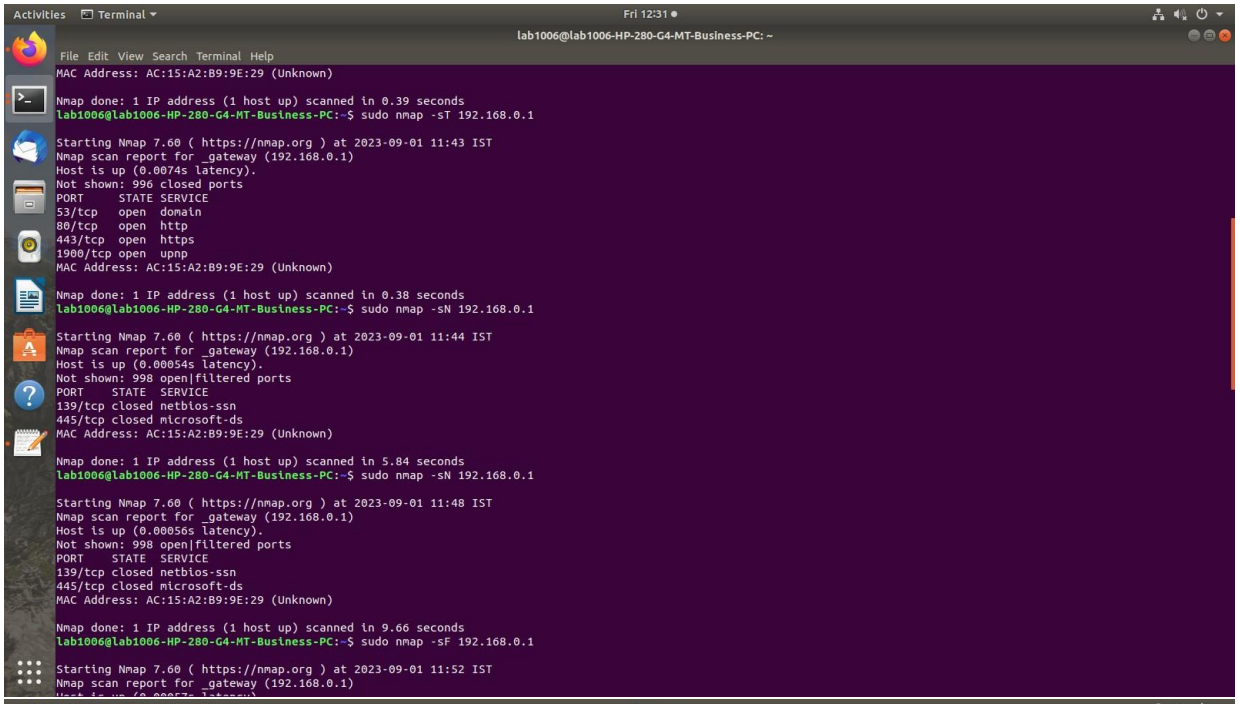
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

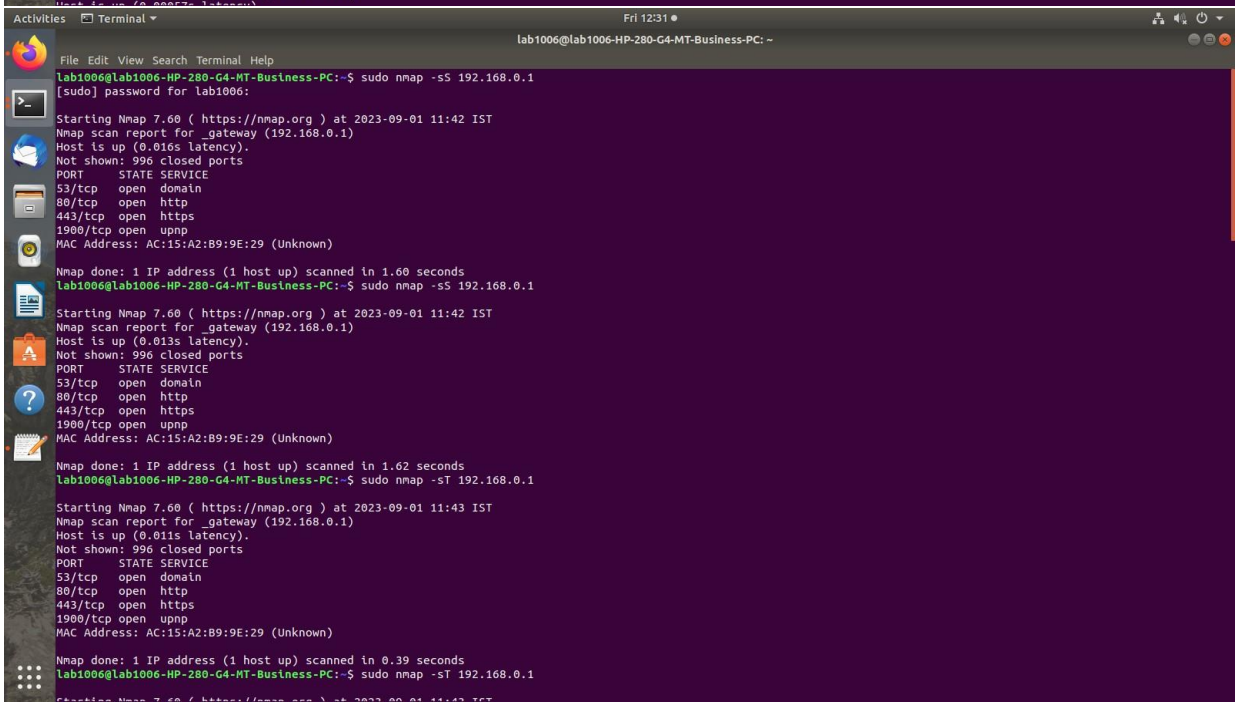




Name : Altaf Alam

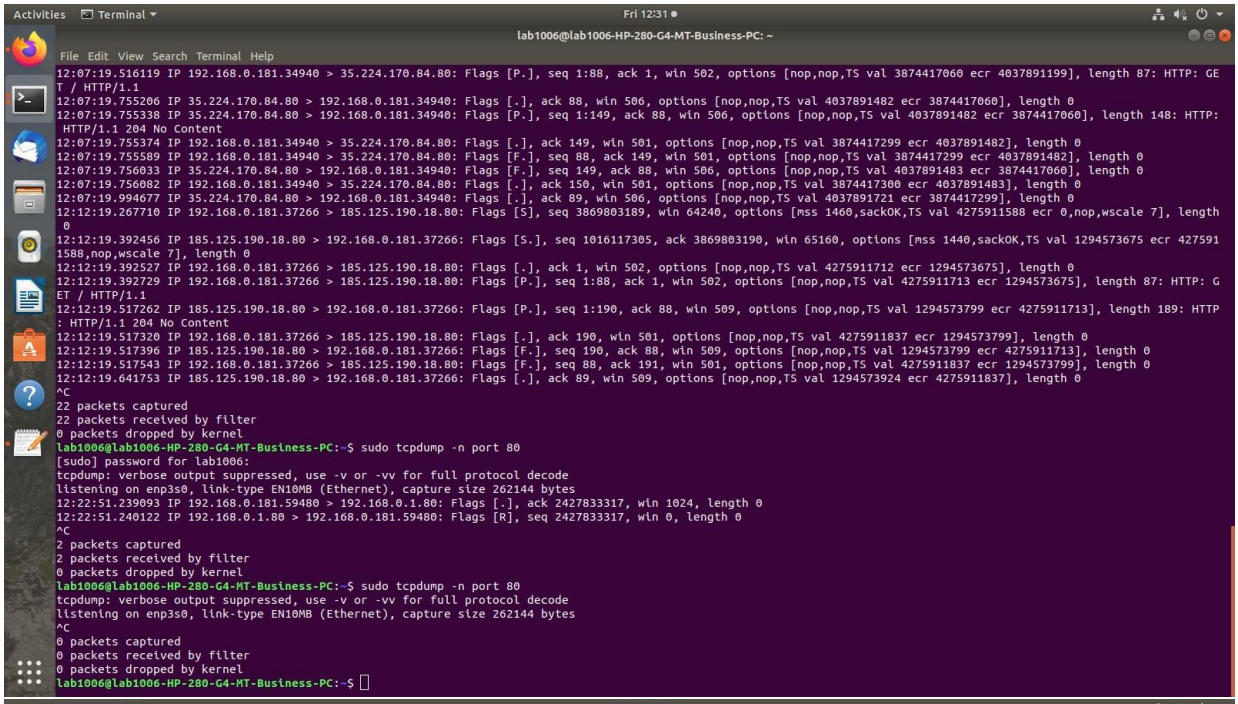
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

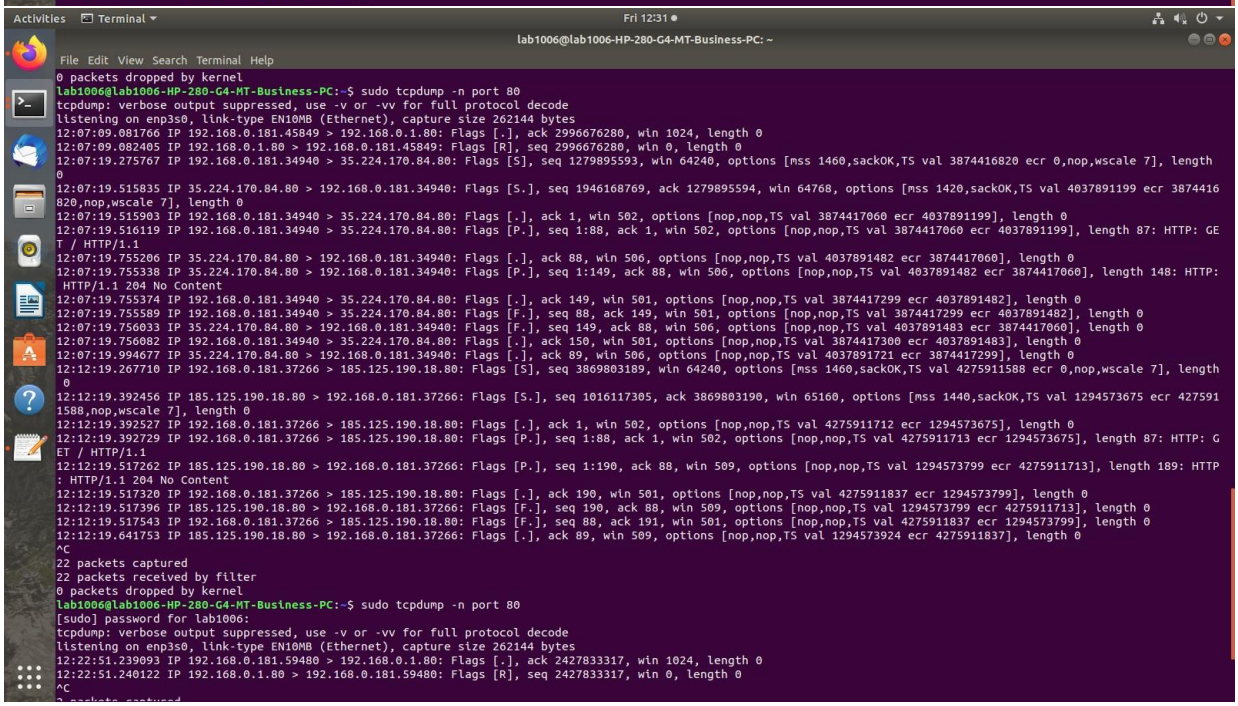




Name : Altaf Alam

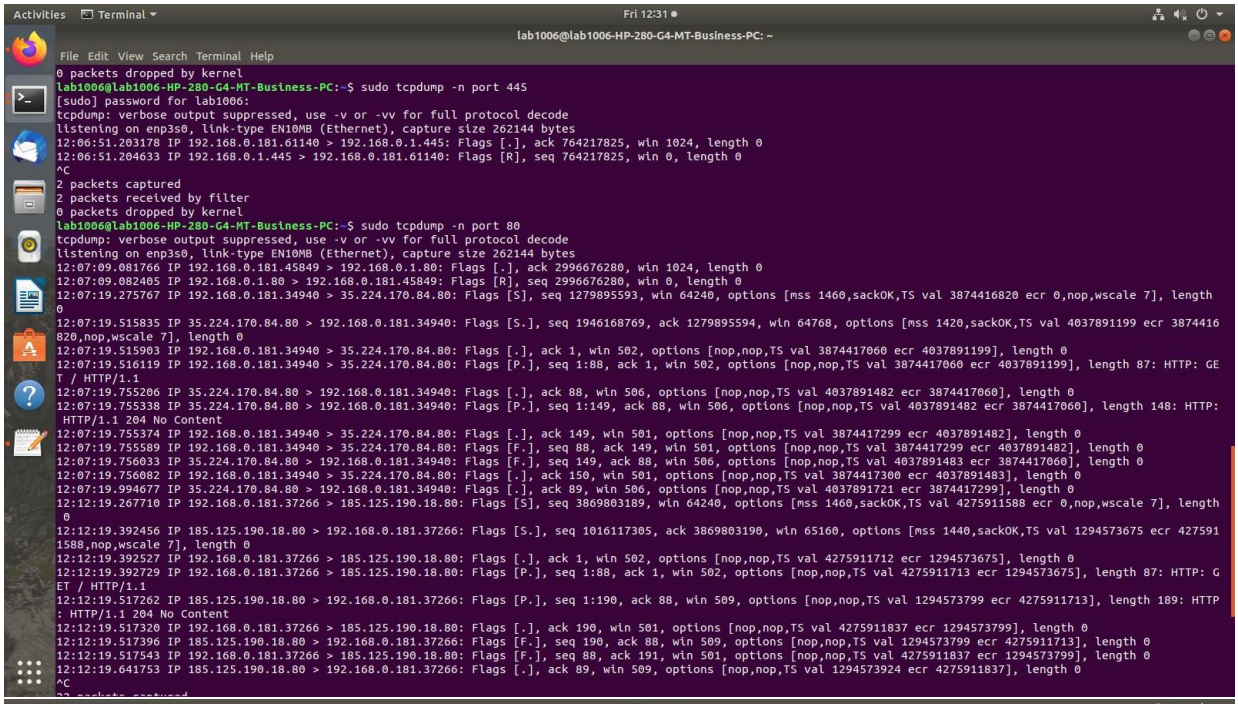
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

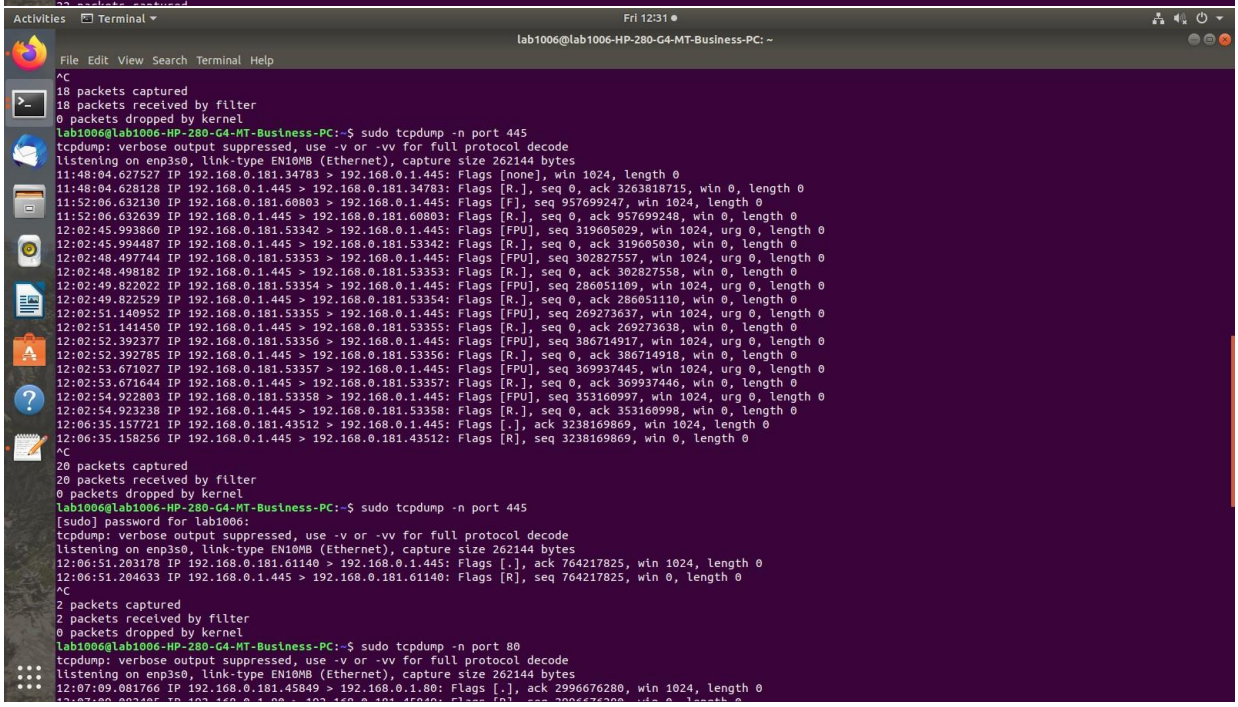




Name : Altaf Alam

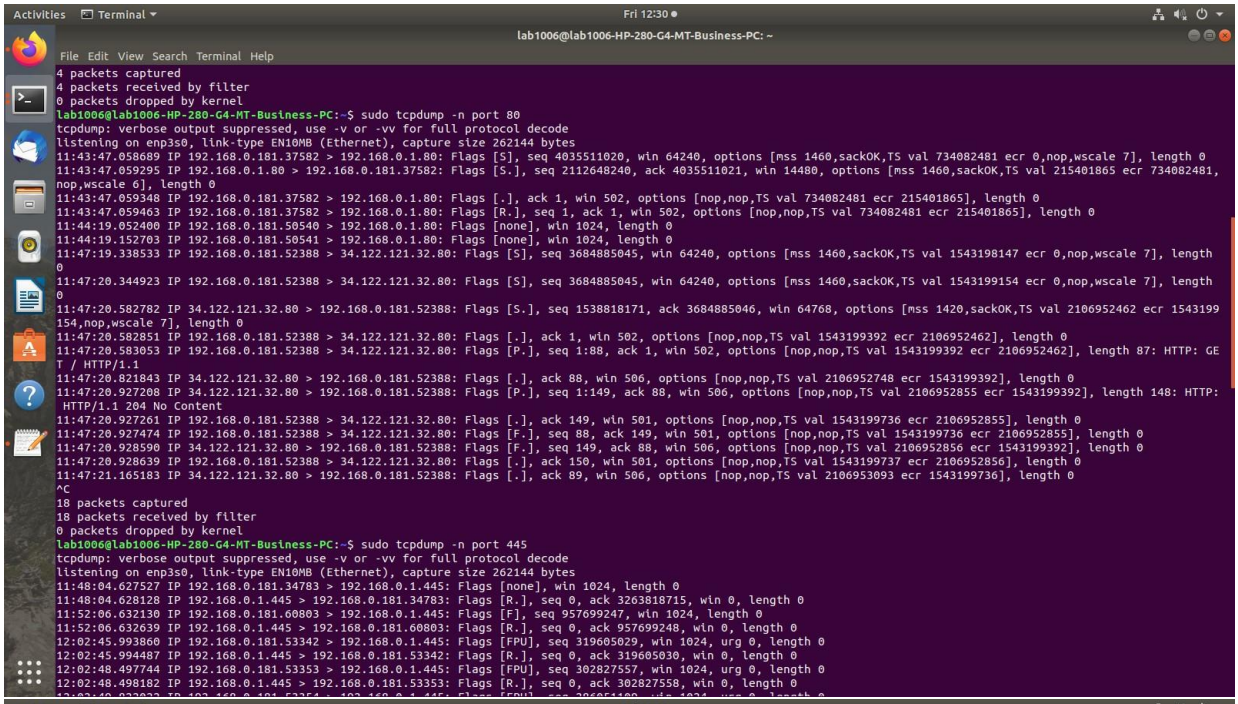
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

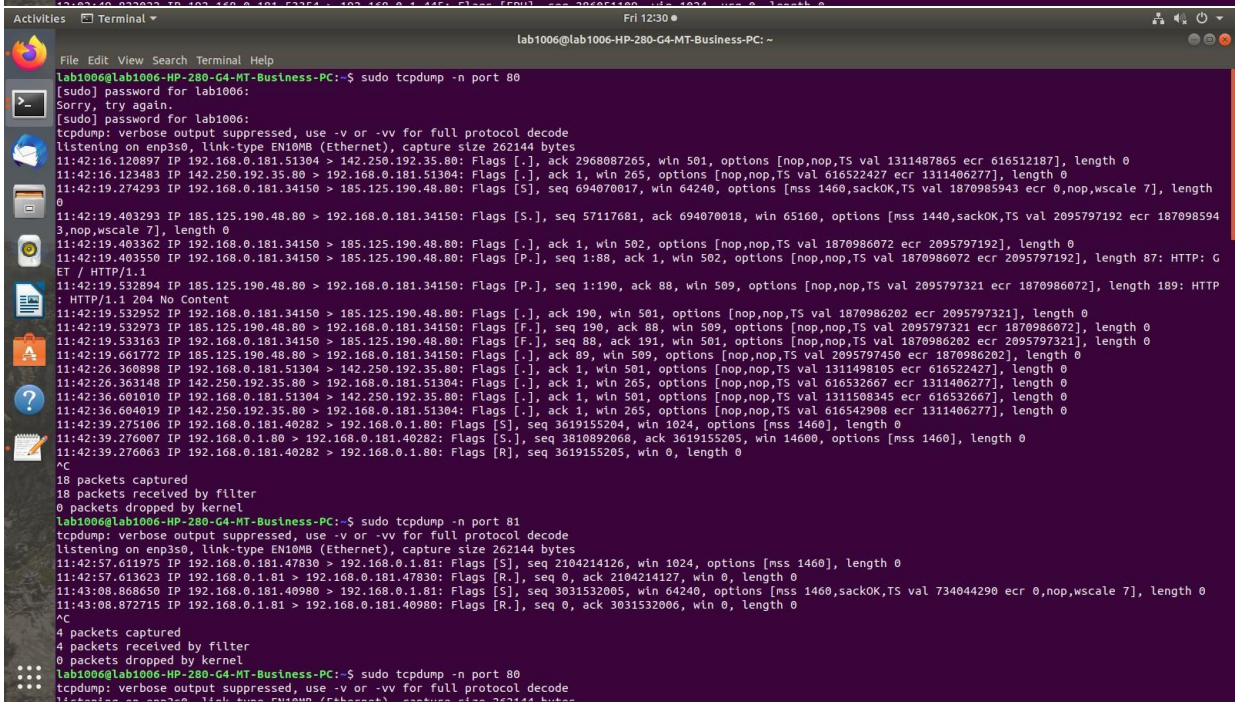




Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23





Conclusion:

In conclusion, Nmap offers a diverse set of scanning techniques to suit various network reconnaissance needs. The choice of scan depends on factors like stealth, speed, and the specific information you seek. Understanding these techniques is vital for network administrators and security professionals to safeguard their networks and systems from potential threats.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

Experiment No 9

Aim : Simulate DOS attack using Hping3.

Lab Outcome :

LO5

Theory :

Understanding Denial of Service (DoS) Attack

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

SYN Flood Attack

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

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

ICMP Flood Attack (Ping Flood Attack)

An ICMP Flood Attack, commonly known as a Ping Flood Attack, targets the Internet Control Message Protocol (ICMP). ICMP is used for various network diagnostic purposes, including the famous "ping" utility, which checks the reachability of a network host. In a Ping Flood Attack,

the attacker sends an overwhelming number of ICMP echo-request packets (ping requests) to a target device.

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

SMURF Attack

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

Here's how a SMURF attack works:

1. The attacker sends a large number of ICMP echo-request packets (pings) to the broadcast address of an intermediate network.

2. The routers on the intermediate network, as per standard behavior, broadcast these ICMP requests to all hosts on the network.

3. Numerous hosts on the intermediate network respond to these ICMP requests by sending ICMP echo-reply packets to the source IP address specified in the requests. Since the source IP address in the requests is the victim's IP address, these responses flood the victim's network.

4. The victim's network becomes overwhelmed with ICMP traffic, leading to high resource utilization and unavailability of services, effectively causing a DoS condition.

Hping3 Commands for SYN Flood and ICMP Flood

SYN Flood using Hping3

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

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

ICMP Flood using Hping3

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

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

Example Hping3 Command for a SMURF Attack

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

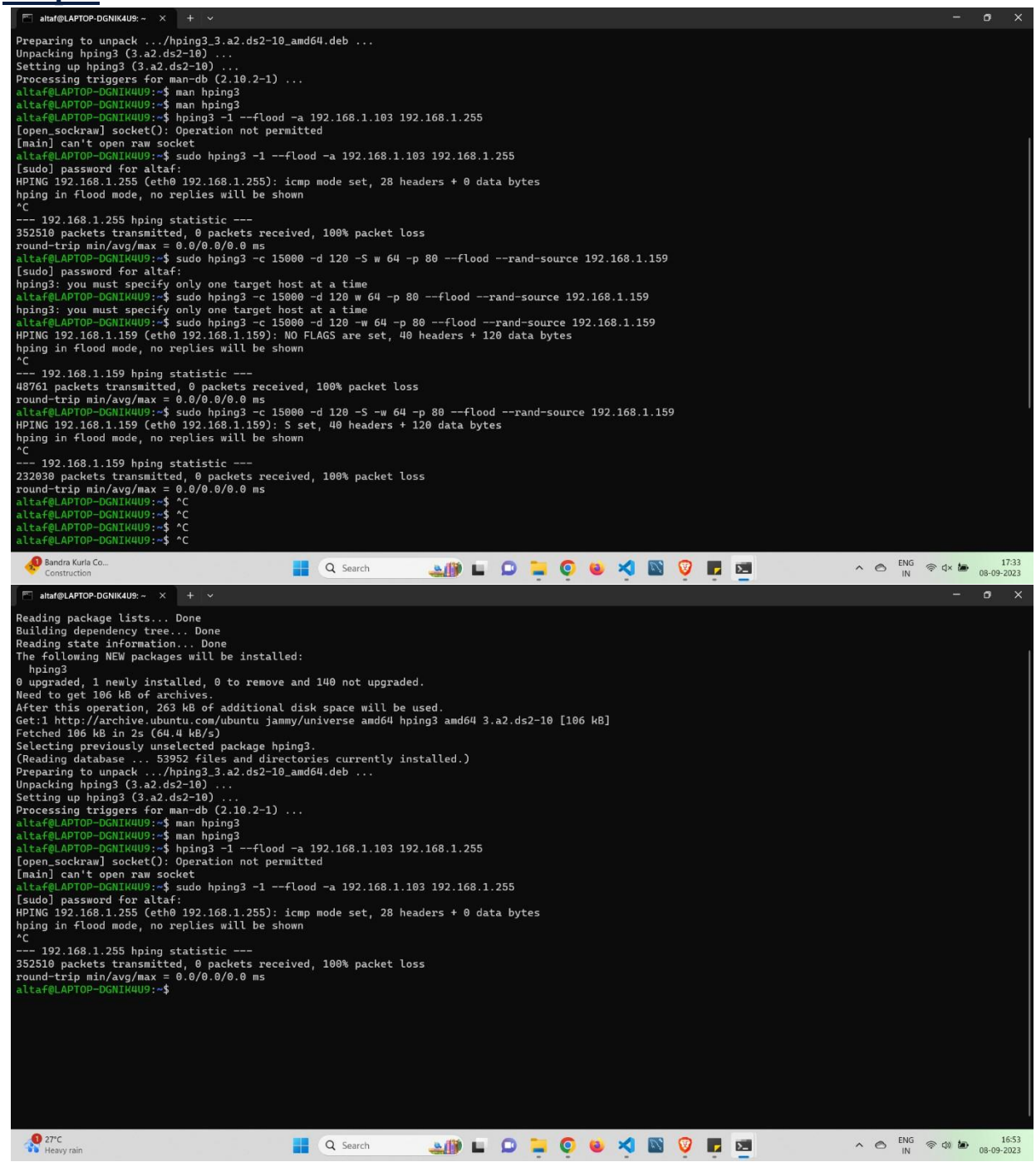
```bash hping3 -1 --flood -a <spoofed_source_ip>
<broadcast_address>
```

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

Output:



Conclusion:

In summary, DoS attacks disrupt network services, with SYN Flood exploiting TCP handshakes and ICMP Flood inundating with ping requests. Effective defense requires awareness, security policies, and intrusion detection.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

Experiment No 10

Aim : To study and configure Firewalls using IP tables

Lab Outcome :

LO6

Theory :

A firewall is a network security device that prevents unauthorized access to a network. It monitors and filters incoming and outgoing traffic based on an organization's security policies.

A firewall can be:

Physical hardware

Digital software

Software as a service (SaaS)

A virtual private cloud

A firewall creates a barrier between a private internal network and the public Internet. It uses a set of security rules to identify and block threats. For example, a firewall rule may say to drop all traffic incoming to port 22, which is commonly used to log in to computers remotely using SSH (secure shell). There are three fundamental types of network firewalls:

1. Packet Filtering (Stateless) Firewalls: These firewalls examine individual packets in isolation and are not aware of the connection state. They can only allow or deny packets based on information in individual packet headers.

2. Stateful Firewalls: Stateful firewalls can determine the connection state of packets, offering more flexibility than stateless firewalls. They collect related packets to determine the connection state before applying firewall rules.

3. Application Layer (Proxy-Based) Firewalls: These go a step further by analyzing the data being transmitted, allowing traffic to be matched against firewall rules specific to individual services or applications.

Regarding the use of IPTables:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

IPTables is a default firewall installed on all official Ubuntu distributions. When you install Ubuntu, IPTables is present but allows all traffic by default. The rules in IPTables are designed to handle three scenarios:

1. Incoming packets destined for your machine (INPUT).
2. Outgoing packets from your machine (OUTPUT).
3. Incoming packets destined for another machine that pass through your machine (FORWARD).

After specifying the traffic type (INPUT, OUTPUT, or FORWARD), you can take three actions:

1. ACCEPT: Allows packets to pass through the firewall.
2. DROP: Ignores the packet and sends no response.
3. REJECT: Ignores the packet but responds to the request with a packet denied message.

You can use various IPTables options, including -A (append rule), -p (connection protocol), -dport (destination port), -j (jump to target), -i (match incoming on a specific interface), -I (insert a rule), -v (display more information), and more to configure and manage firewall rules.

Additionally, you can allow incoming traffic on specific ports, block unwanted traffic, and even configure rules for specific source machines. The rules are divided into tables such as Filter, NAT, Mangle, and Raw, each serving specific functions in network traffic management.

Basic commands

sudo iptables -L

(-L - List the current filter rules.)

As you can see, we have our three default chains (INPUT,OUTPUT, and FORWARD). We also can see each chain's default policy (each chain has ACCEPT as its default policy). We also see some column headers, but we don't see any actual rules. This is because Ubuntu doesn't ship with a default rule set.

Basic Iptables Options

- `-A` - Append this rule to a rule chain. Valid chains for what we're doing are INPUT,

FORWARD and OUTPUT, but we mostly deal with INPUT in this tutorial, which affects only incoming traffic.

- `-p` - The connection protocol used.
- `--dport` - The destination port(s) required for this rule. A single port may be given, or a range may be given as `start:end`, which will match all ports from `start` to `end`, inclusive.
- `-j` - Jump to the specified target. By default, iptables allows four targets:
- `ACCEPT` - Accept the packet and stop processing rules in this chain.
- `REJECT` - Reject the packet and notify the sender that we did so, and stop processing rules in this chain.
- `DROP` - Silently ignore the packet, and stop processing rules in this chain.
- `LOG` - Log the packet, and continue processing more rules in this chain.

  Allows the use of the `--log-prefix` and `--log-level` options.
- `-i` - Only match if the packet is coming in on the specified interface.
- `-I` - Inserts a rule. Takes two options, the chain to insert the rule into, and the rule number it should be.
- `-I INPUT 5` would insert the rule into the INPUT chain and make it the 5th rule in the list.
- `-v` - Display more information in the output. Useful for if you have rules that look similar without using `-v`.
- `-s --source` - address[/mask] source specification
- `-d --destination` - address[/mask] destination specification

- `-o --out-interface` - output name[+] network interface name ([+] for wildcard)

Allowing Incoming Traffic on Specific Ports

You could start by blocking traffic, but you might be working over SSH, where you would need to allow SSH before blocking everything else.

To allow incoming traffic on the default SSH port (22), you could tell iptables to allow all TCP traffic on that port to come in.

```
sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Referring back to the list above, you can see that this tells iptables:

- append this rule to the input chain (-A INPUT) so we look at incoming traffic
- check to see if it is TCP (-p tcp).
- if so, check to see if the input goes to the SSH port (--dport ssh).
- if so, accept the input (-j ACCEPT).

Now, let's allow all incoming web traffic

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

We have specifically allowed tcp traffic to the ssh and web ports, but as we have not blocked anything, all traffic can still come in.

Blocking Traffic

Once a decision is made to accept a packet, no more rules affect it. As our rules allowing ssh and web traffic come first, as long as our rule to block all traffic comes after them, we can still accept the traffic we want. All we need to do is put the rule to block all traffic at the end.

```
sudo iptables -A INPUT -j DROP
```

```
sudo iptables -L
```

Because we didn't specify an interface or a protocol, any traffic for any port on any interface is blocked, except for web and ssh.

Editing iptables

The only problem with our setup so far is that even the loopback port is blocked. We could have written the drop rule for just eth0 by specifying -i eth0, but we could also add a rule for the loopback. If we append this rule, it will come too late - after all the traffic has been dropped. We need to insert this rule before that. Since this is a lot of traffic, we'll insert it as the first rule so it's processed first.

```
sudo iptables -I INPUT 1 -i lo -j ACCEPT sudo  iptables
-L
```

we will list iptables in greater detail.

```
sudo iptables -L -v
```

You can now see a lot more information. This rule is actually very important, since many programs use the loopback interface to communicate with each other.

Allow traffic on ICMP port sudo iptables -A

INPUT -p icmp -j ACCEPT now list the rules

again.. sudo iptables -L

clearing all rules sudo

iptables -F sudo iptables

-L

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

Dropping icmp packets try to ping ur

neighbour machine ping 192.168.92.17 u

can see the response packets received.

Now block incoming icmp packets from the neighbour using

command: sudo iptables -A **INPUT** -p icmp -j DROP list the rule: sudo

iptables -L try to ping ur neighbour machine again ping 192.168.92.17

u can not see receive icmp echo reply packets..

Now try to restrict outgoing icmp packets by adding rule sudo

iptables -A **OUTPUT** -p icmp -j DROP

List the rule: sudo iptables L

now try to ping neighbour

ping 192.168.92.17

flush all rules and try to ping neighbor

Blocking TCP port traffic will not allow u to browse the Internet

sudo iptables -A INPUT -p tcp -j DROP List the rule: sudo iptables -

L

Now try to access the internet.. u can,t. Flush the rule n then try to acess internet.. u can.

Blocking ICMP packets from specific source machine:

sudo iptables -A INPUT -s 192.168.92.17 -p icmp -j DROP

sudo iptables -L ping 192.168.92.17

---does not allow (192.168.92.17 can not send u icmp

packets) ping any other machine: ping 192.168.92.11 --

allowed (192.168.92.11 can send u icmp packets)

Types of iptables:

I. IPTABLES TABLES and CHAINS

IPTables has the following 4 built-in tables.

1. Filter Table

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

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

  iptables -t filter -L

2. NAT table

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

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

  command and see the result sudo iptables -t nat -L

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

3. Mangle table

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

- PREROUTING chain
- OUTPUT chain
- FORWARD chain
- INPUT chain
- POSTROUTING chain

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

4. Raw table

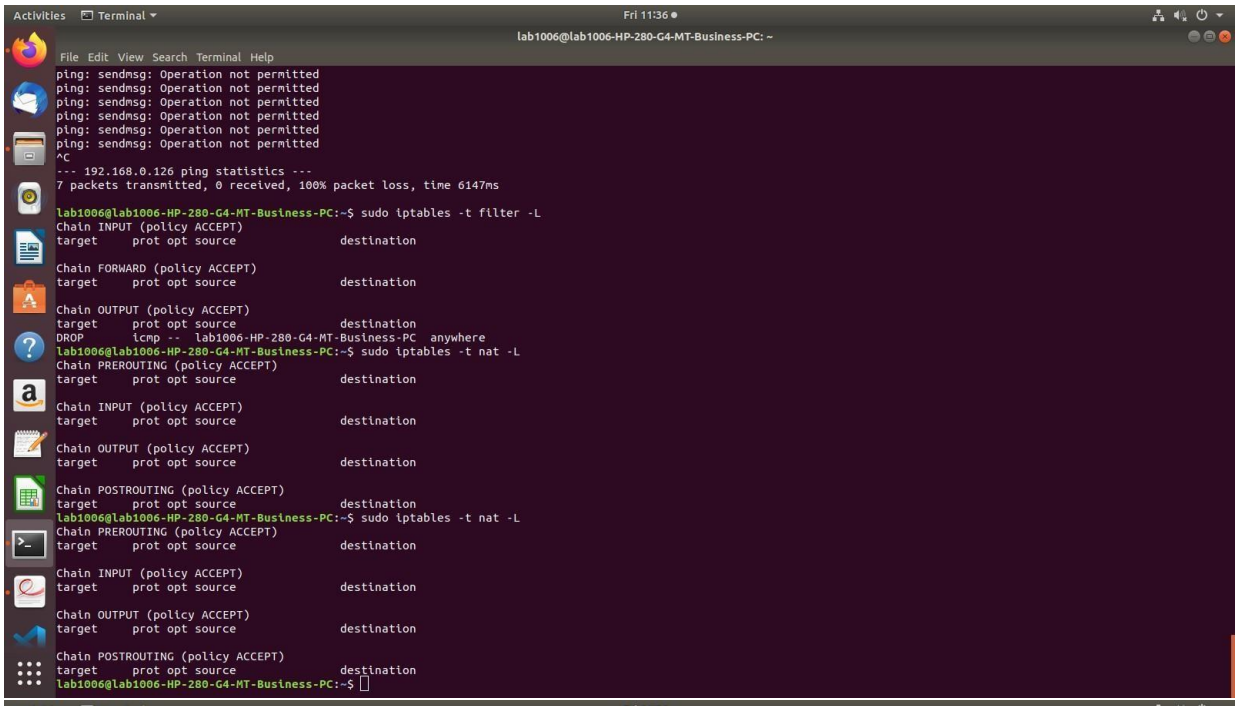
Iptable's Raw table is for configuration exemptions. Raw table has the following built-in chains.

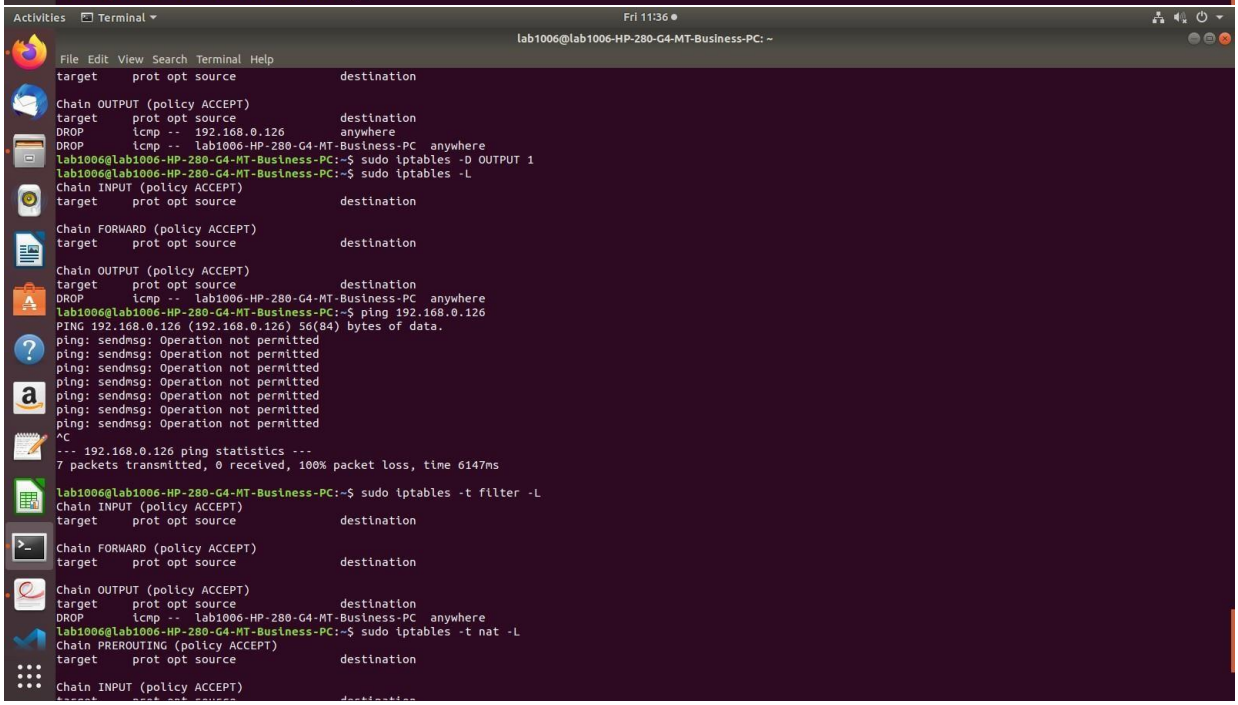
- PREROUTING chain
- OUTPUT chain

Output:

Name : Altaf Alam

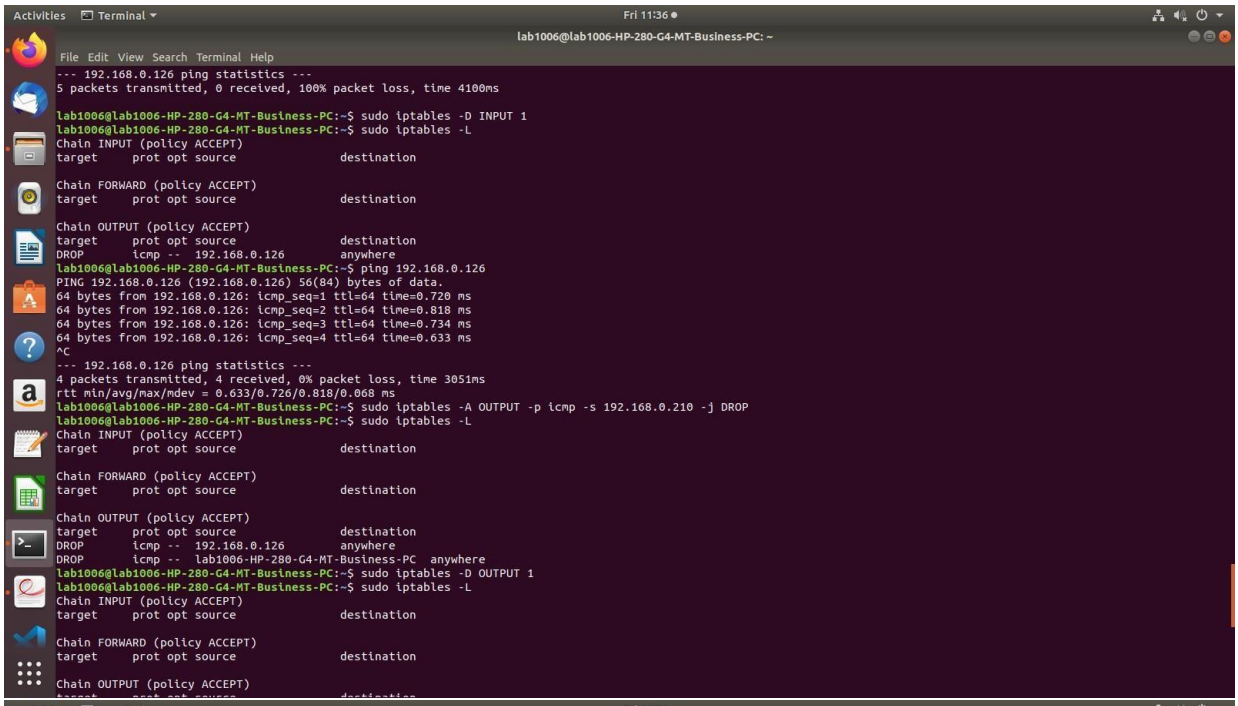
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

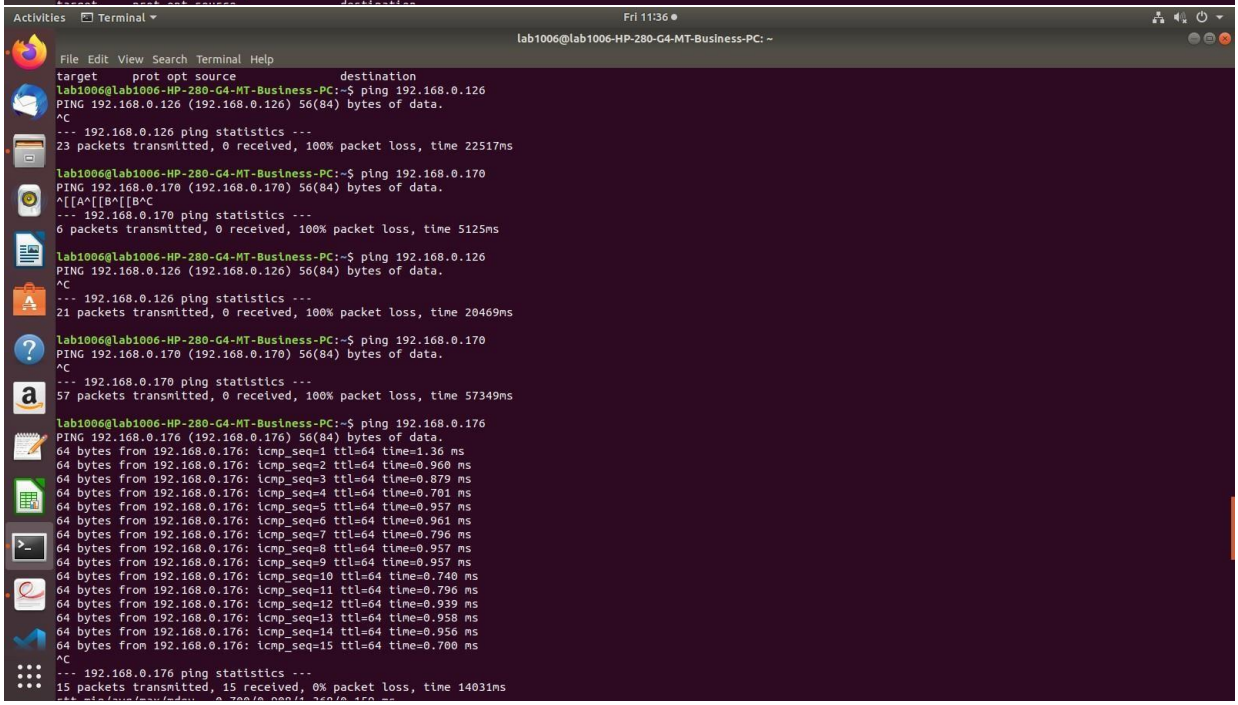




Name : Altaf Alam

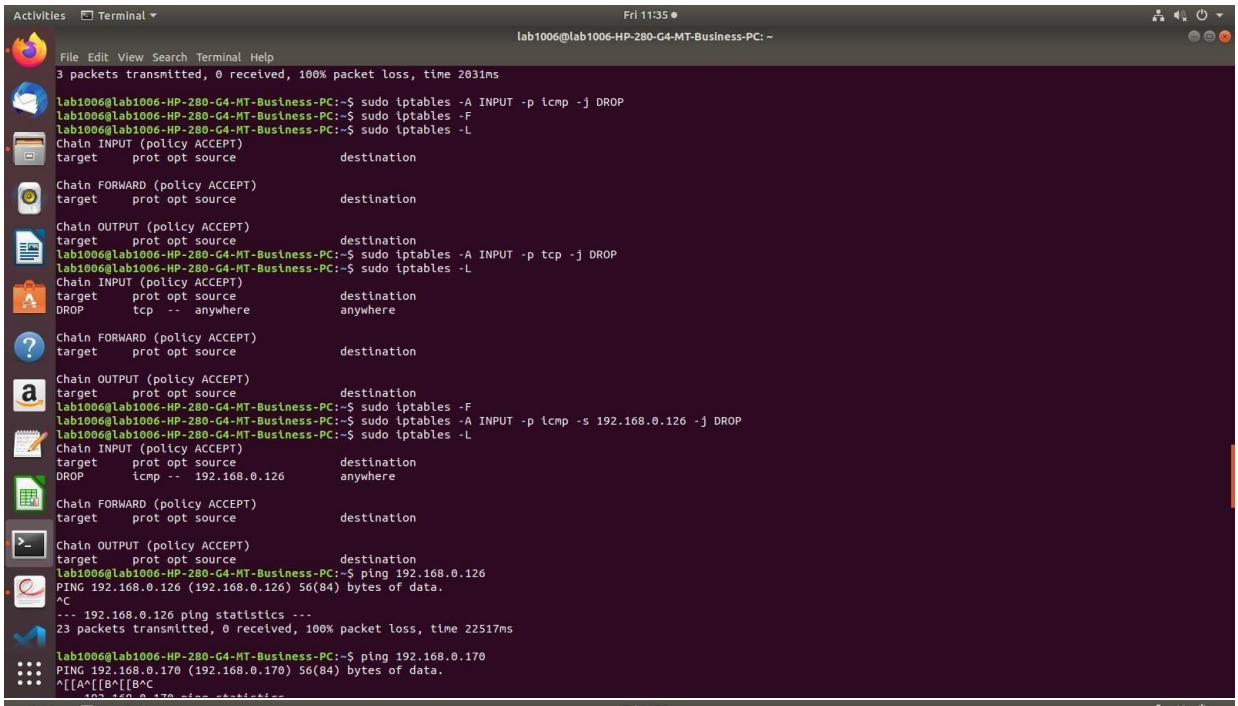
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

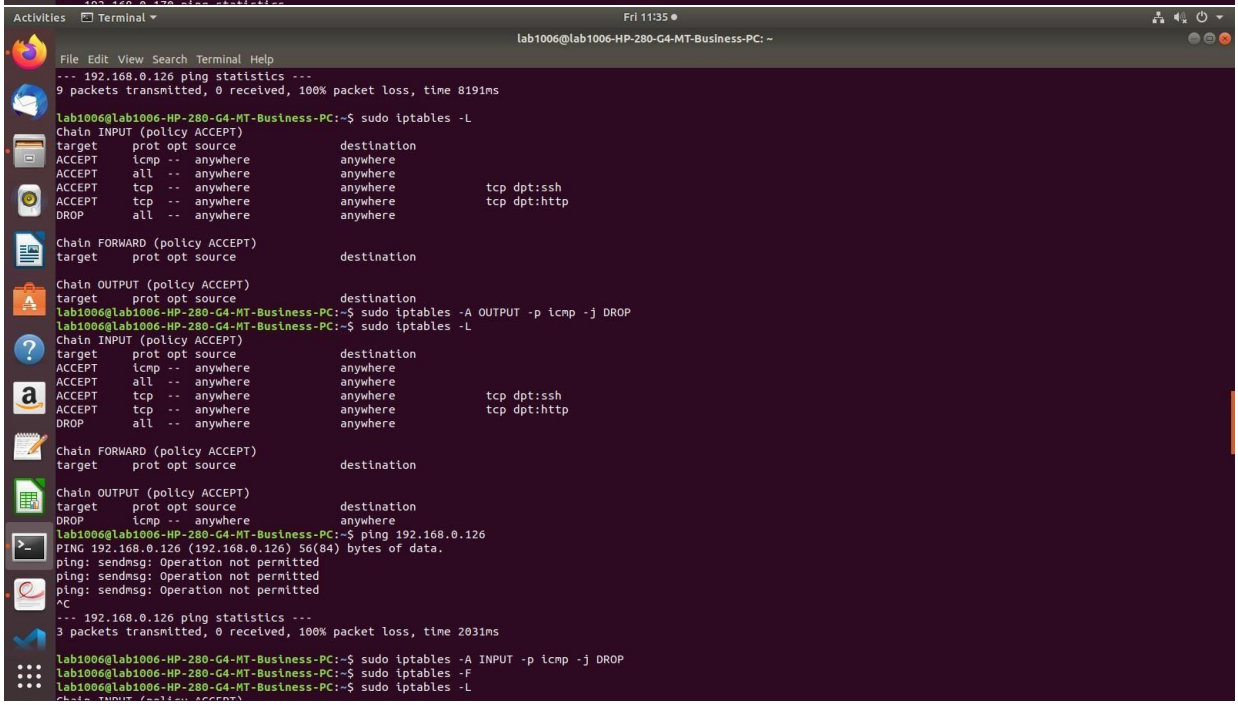




Name : Altaf Alam

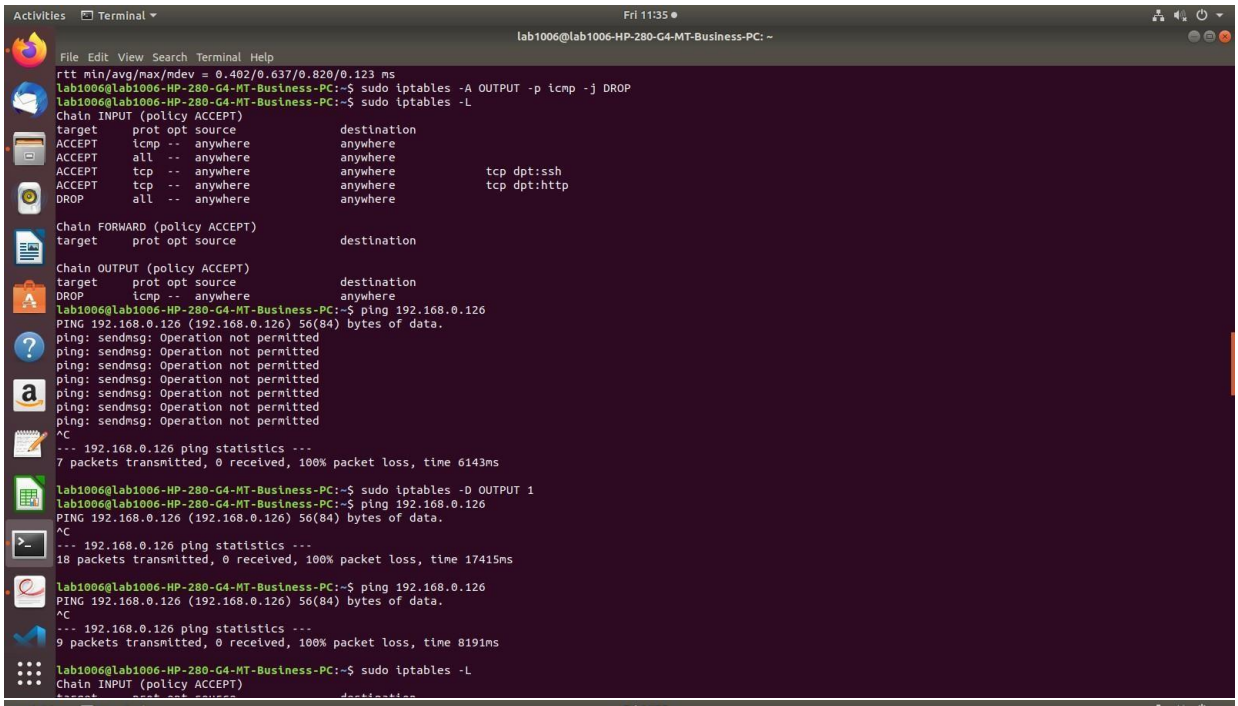
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

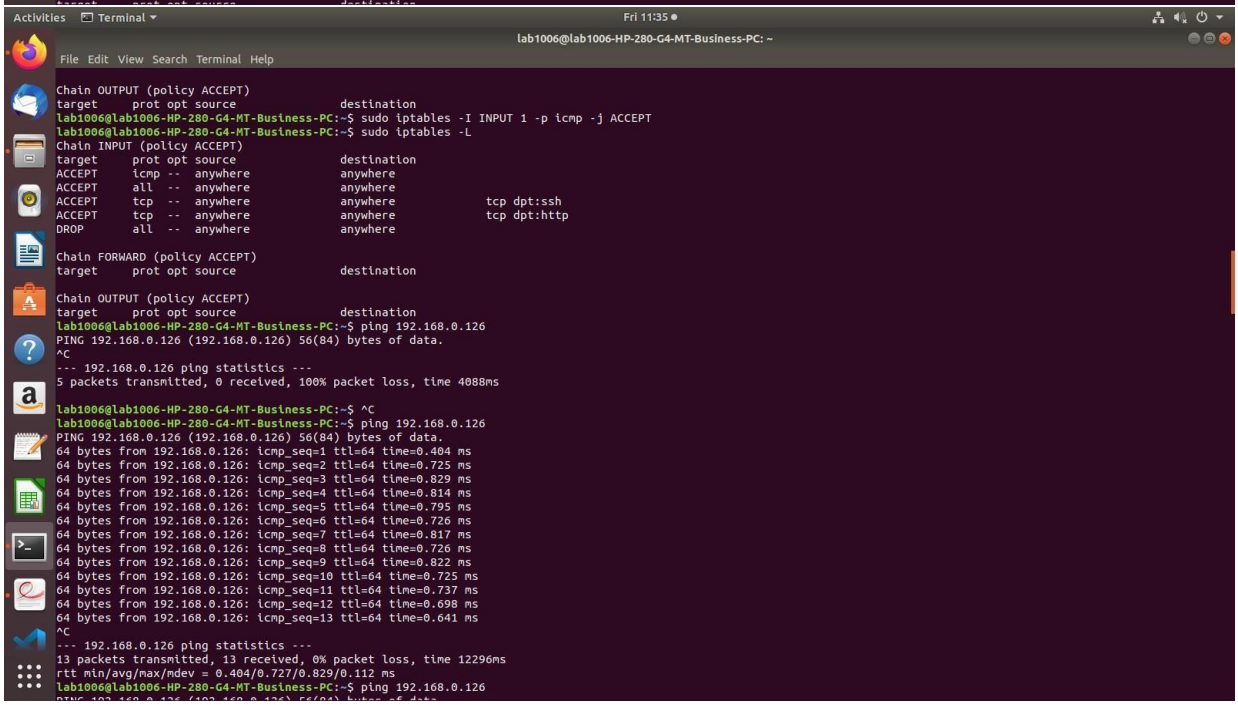




Name : Altaf Alam

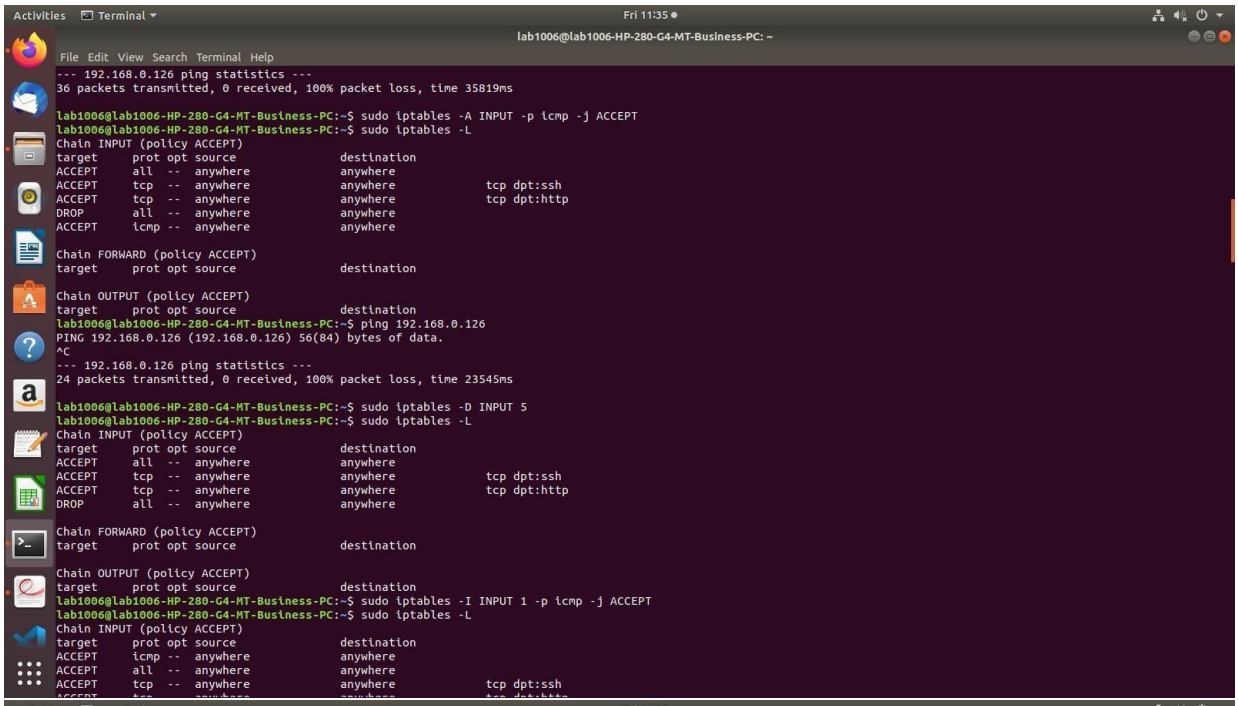
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

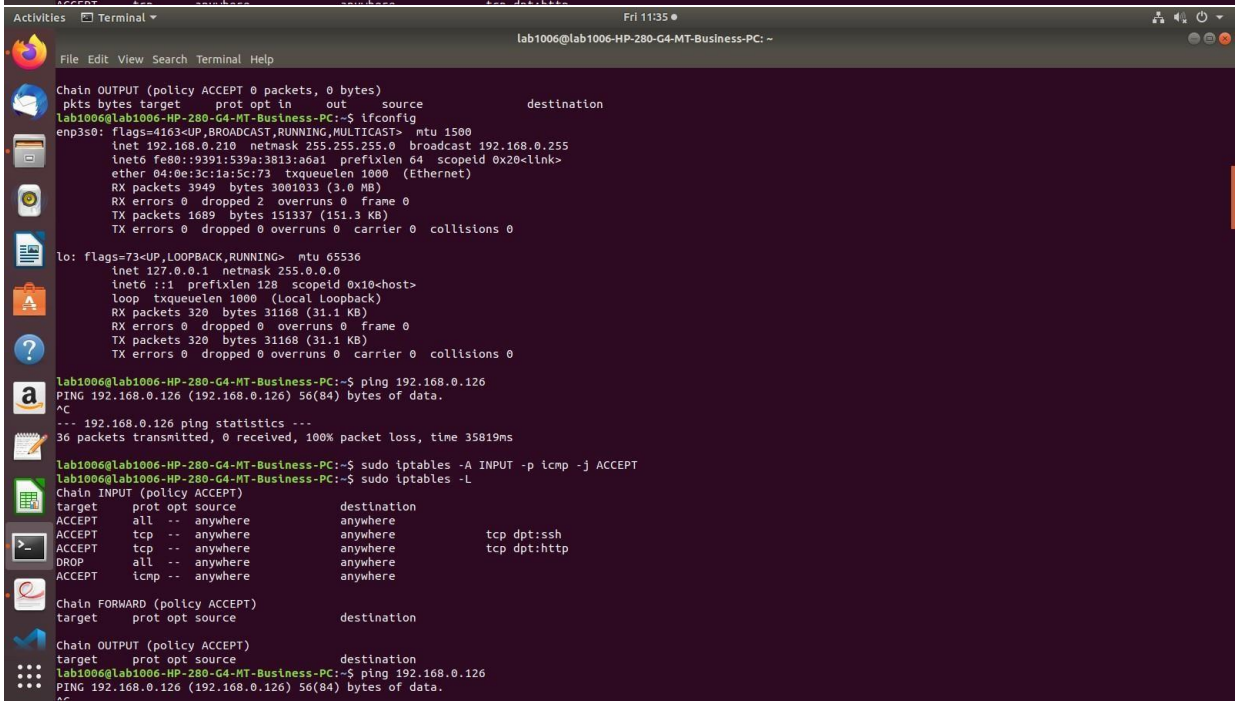




Name : Altaf Alam

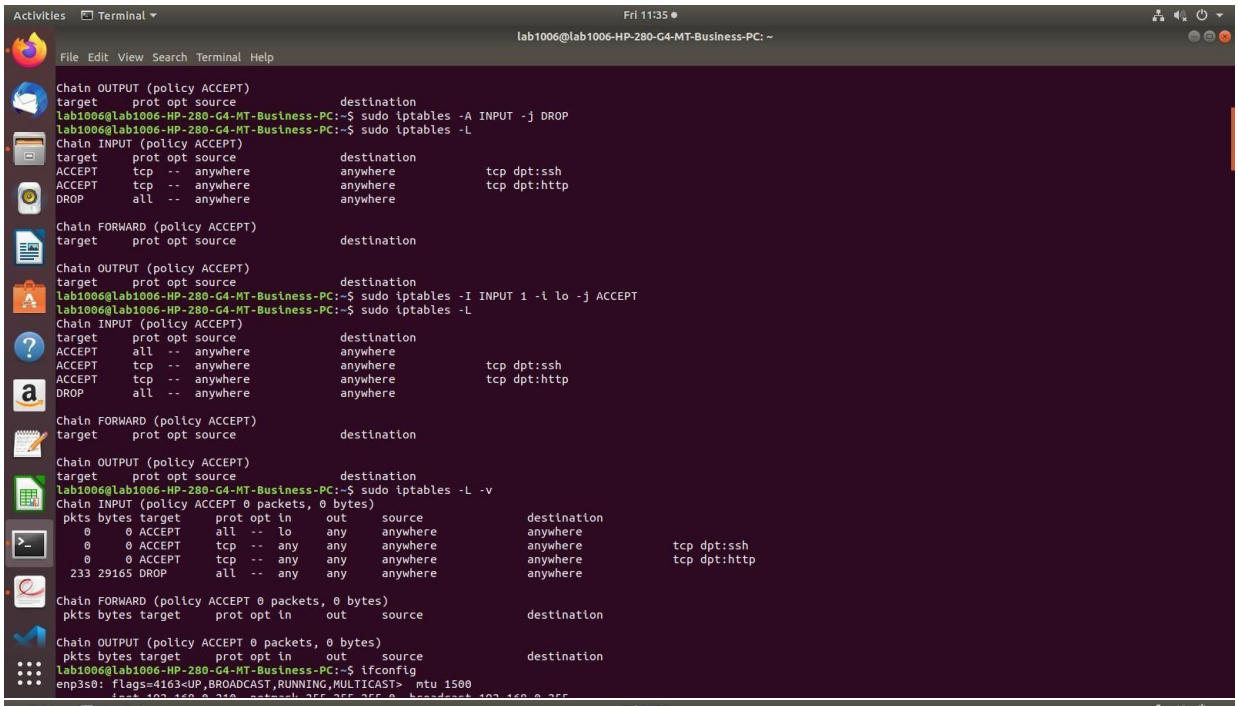
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23

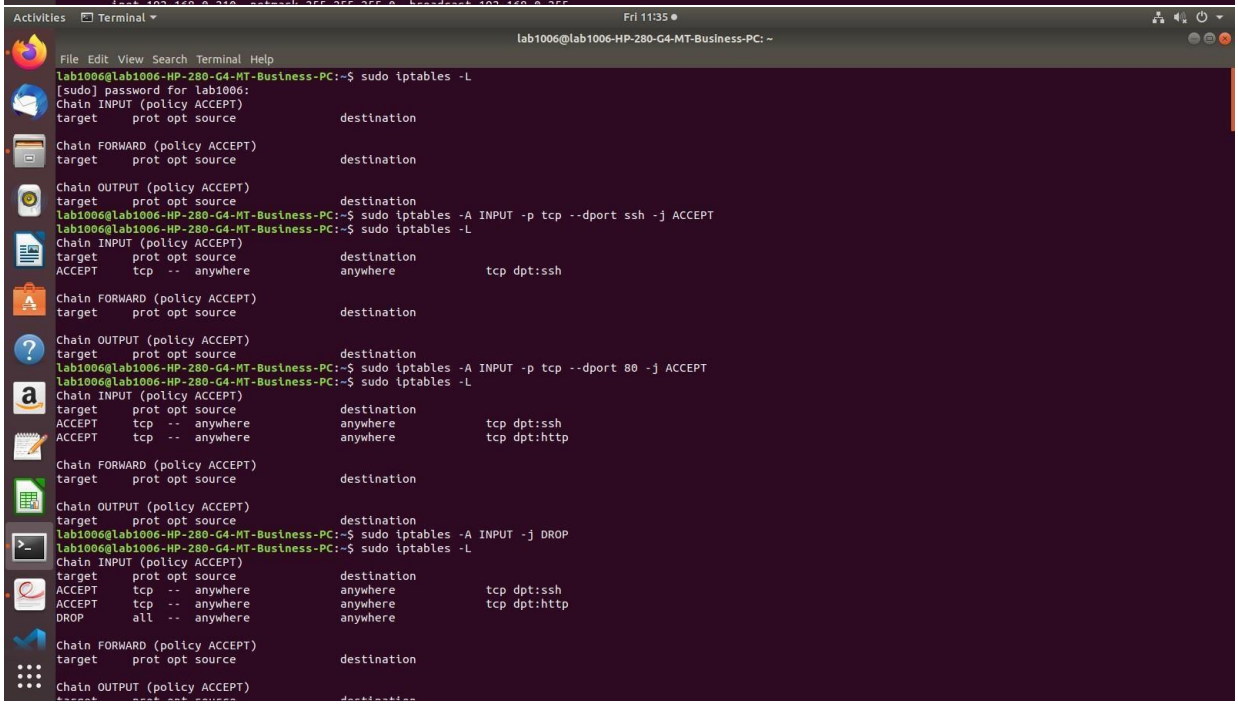




Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 07/09/23





Conclusion:

In conclusion, we studied about the firewalls using iptables and configured it. We also learn to implement various commands.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 25/08/23

Experiment No 11

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

Lab Outcome :

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

Theory :

1. What is Intrusion Detection System?

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

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

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

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

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

1. Sniffer Mode: In this mode, Snort behaves like a packet sniffer, capturing and displaying network traffic in real-time. It is valuable for troubleshooting network issues and

gaining insights into the flow of data. However, it doesn't involve analysis or rule-based detection.

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

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

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

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

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

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

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

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

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

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

5. Make the following changes in the configuration file: - In section 1, set:
```

```
ipvar HOME_NET 192.168.44.0/24
```

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

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

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

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

10. Observe the output in the Snort terminal.

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

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

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 25/08/23

```

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

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

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

i. Compare alerts generated using different rules.

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

## **Output**:

**Conclusion:**

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

Experiment No 12

Aim : Explore the GPG Tool of linux to implement email security.

Lab Outcome :

LO6

Theory :

1. What is Private Key Ring and Public Key Ring?

In the context of GPG (GNU Privacy Guard), a private key ring and a public key ring are essential components of the OpenPGP encryption and signing system. These key rings are used for managing cryptographic keys for secure communication.

- Private Key Ring: This is a collection of private keys owned by a user. Private keys are used for decrypting messages sent to you and for signing messages to ensure their authenticity. Each user typically has their private key ring, which should be kept confidential and protected at all costs. Only the owner of the private key ring should have access to it.

- Public Key Ring: This is a collection of public keys, which are meant to be shared openly. Public keys are used by others to encrypt messages meant for you and to verify the digital signatures you create with your private key. Public keys are freely distributed and can be obtained from a keyserver or directly from the person they belong to.

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

Key Generation
To generate private and public key pairs for sender and receiver, you can use the following commands:

```bash
gpg --gen-key   or gpg --full-generate-key (repeat for sender and receiver)
```

Exporting and Importing Keys
- Create a file containing sender's public key (ASCII format):
```bash
gpg --export -a username > filename
```
- Create a file containing sender's private key:
```bash
```

gpg --export-secret-key -a username > filename
```
- Import the public key of the receiver:
```bash
gpg --import filename_containing_public_key_of_receiver
```

Signing Keys
Sender can sign the public key of the receiver to establish trust:
```bash
gpg --sign-key receiver_email
```

Encrypting Data
Encrypt a file for a specific receiver:
```bash
gpg --encrypt -r receiver_email name_of_file  .gpg file created
```

Encrypt and sign a file (ASCII format):
```bash
gpg --encrypt --sign --armor -r receiver_email name_of_file  ASCII file created
```

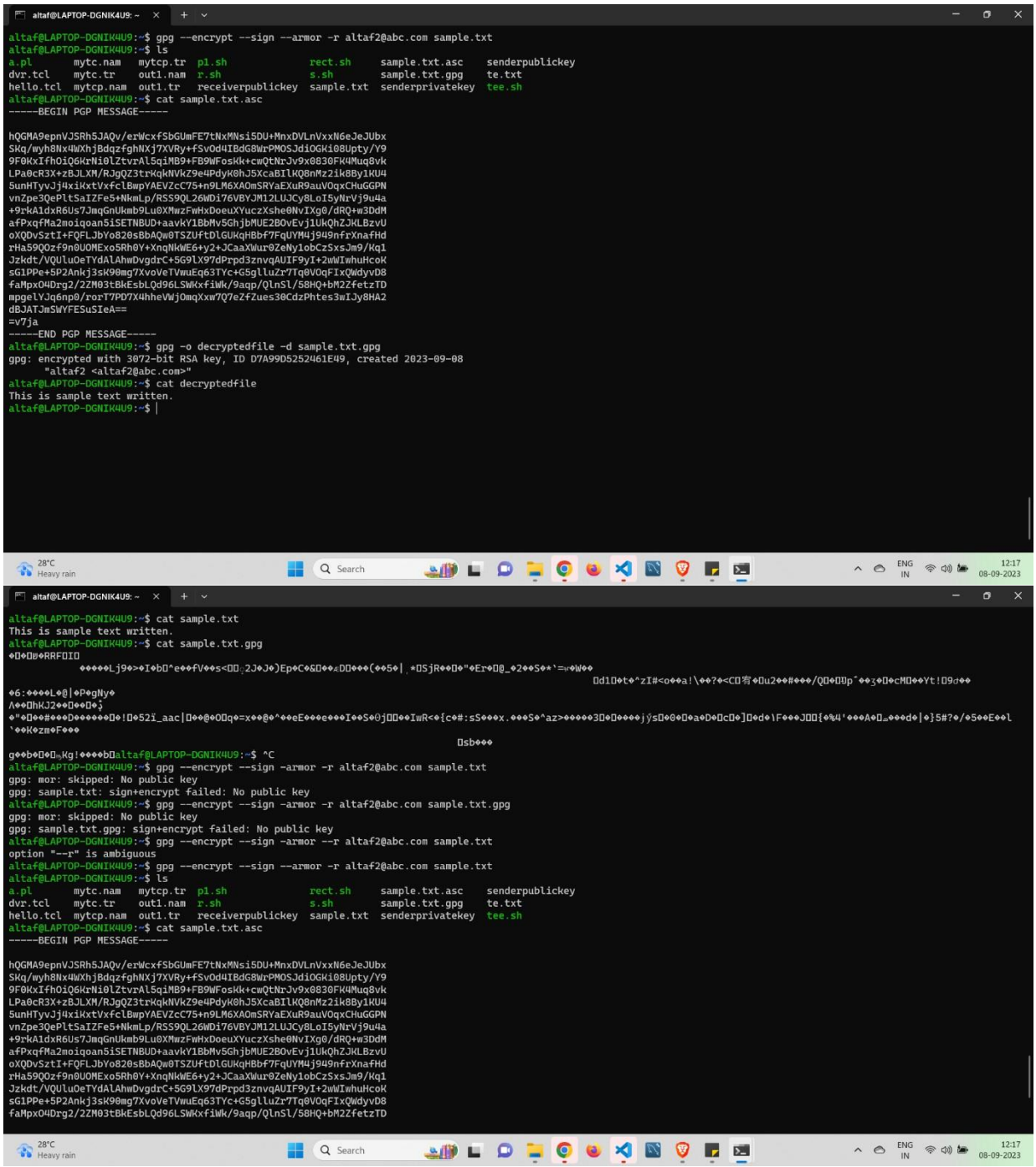
Encrypt and sign a file (.gpg format):
```bash
gpg --encrypt --sign -r receiver_email name_of_file  .gpg file created
```

Decrypting Data
Decrypt a file:
```bash
gpg -o myfiledecrypted -d myfile.txt.gpg
```
```

Name : Altaf Alam

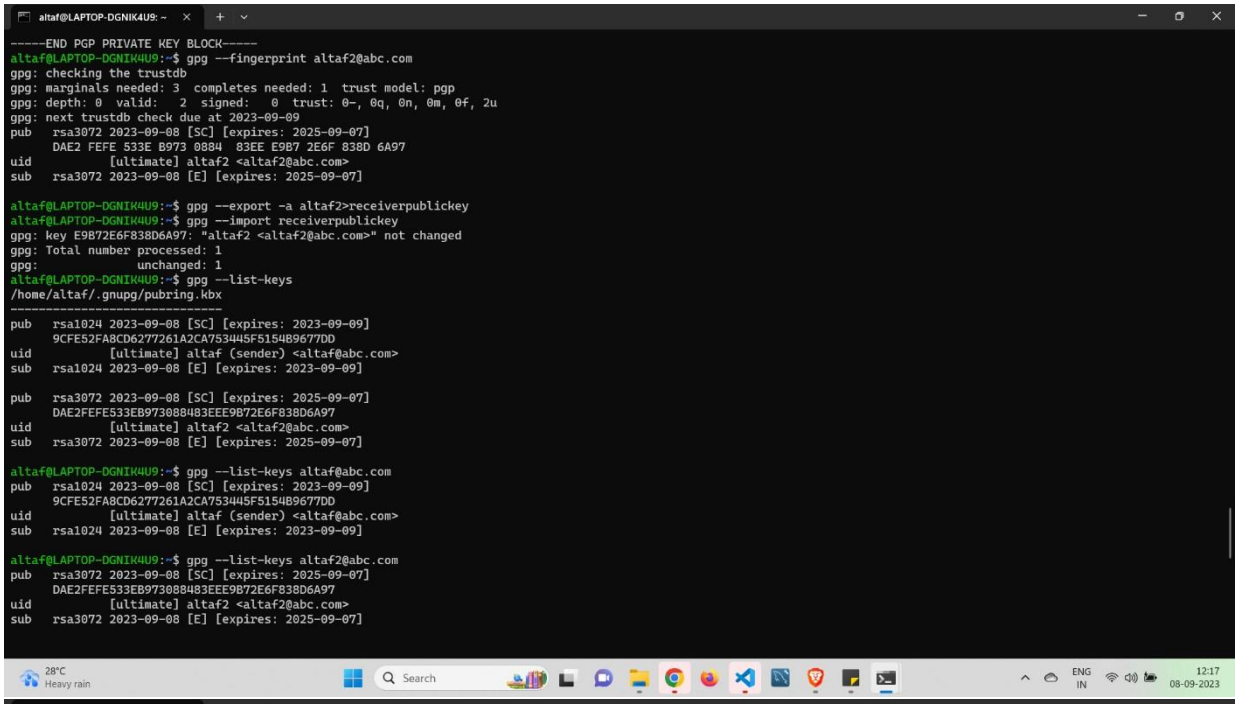
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

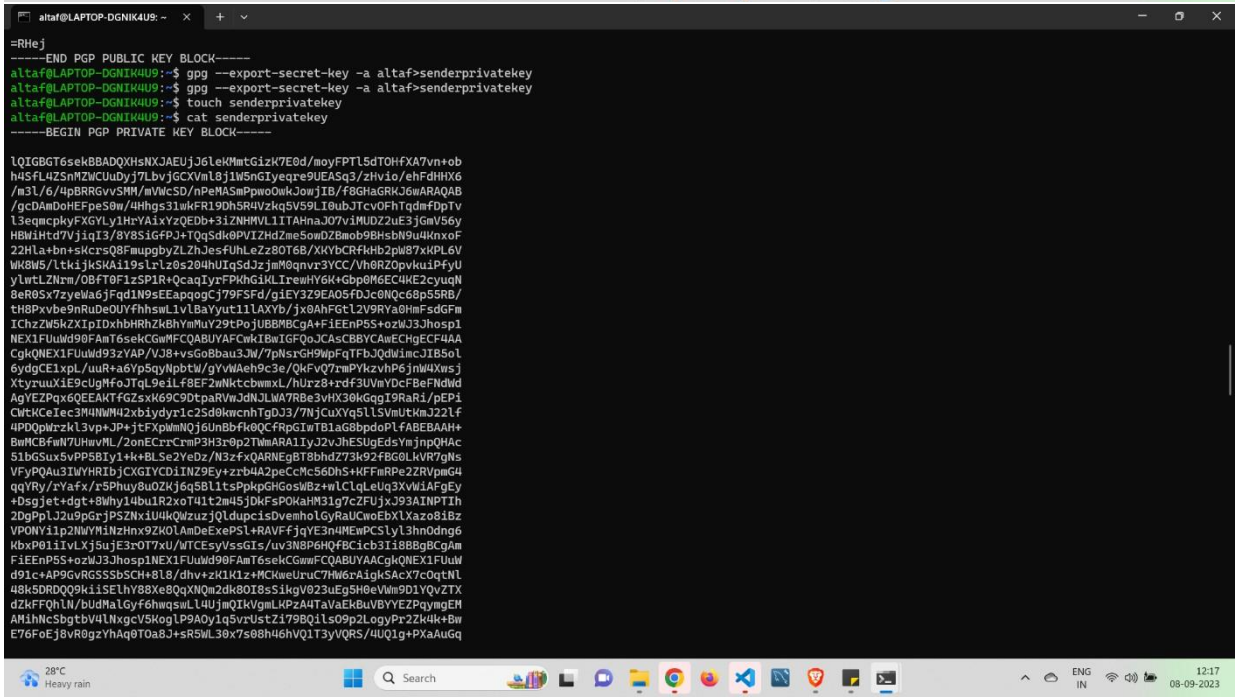
Output:



Name : Altaf Alam

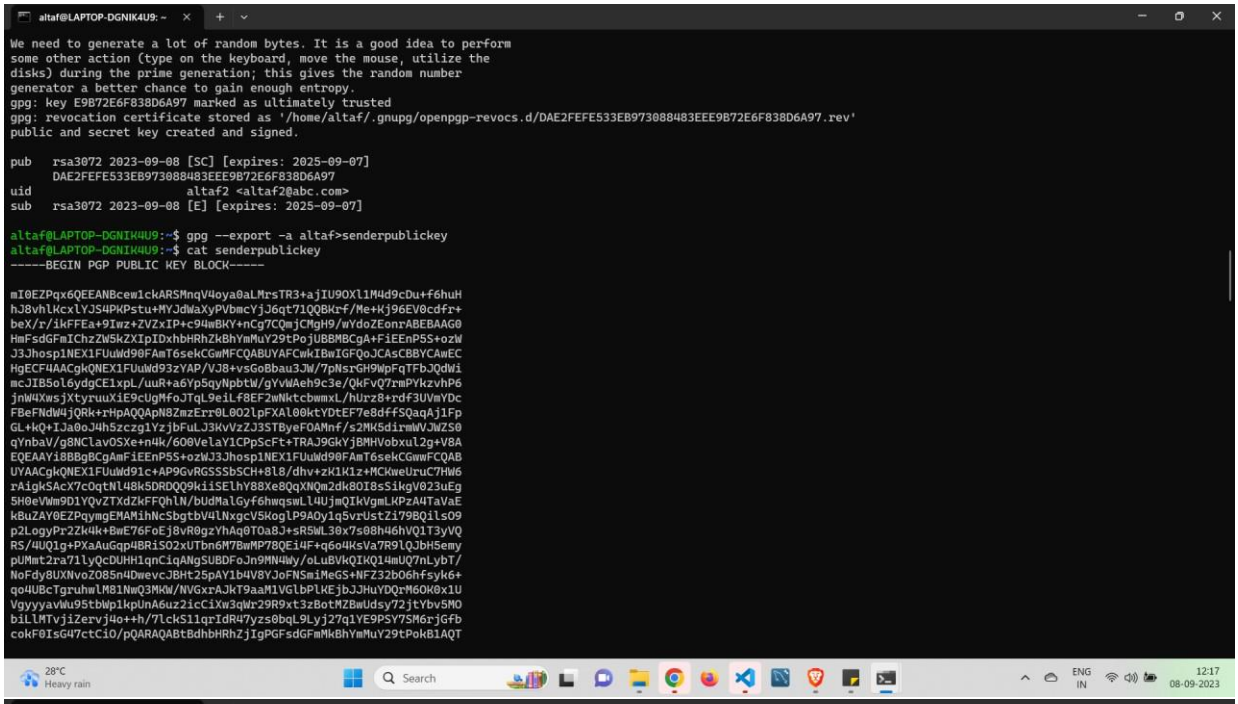
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

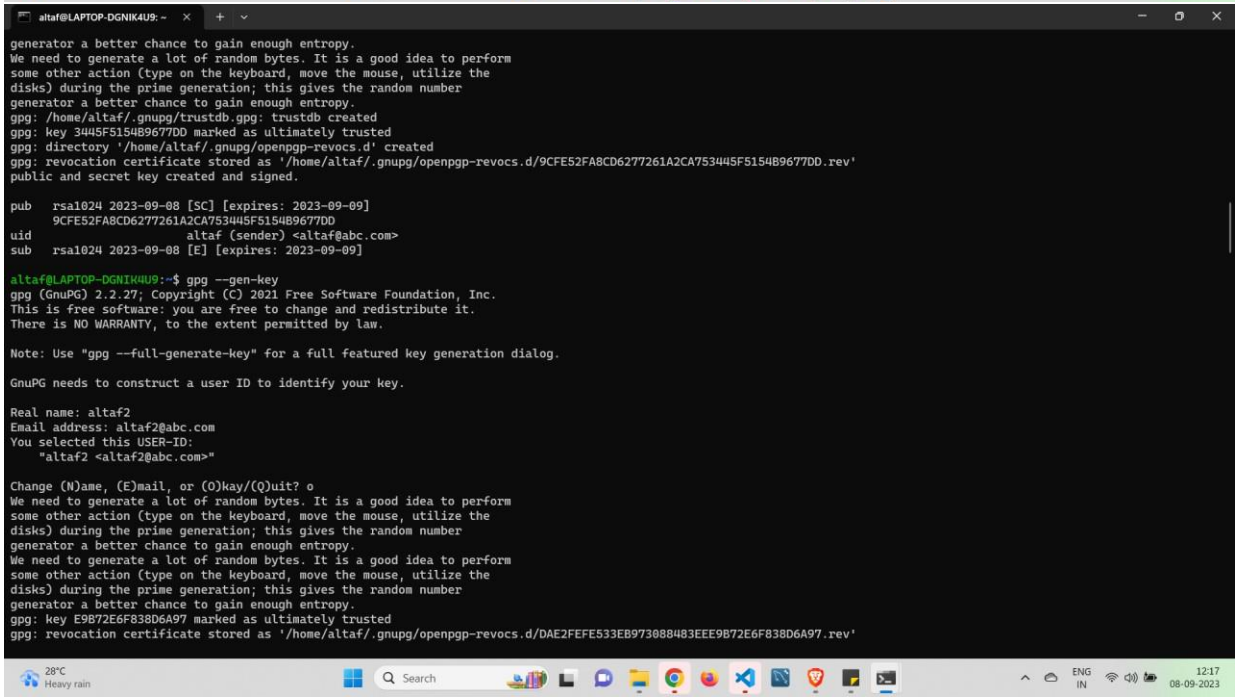




Name : Altaf Alam

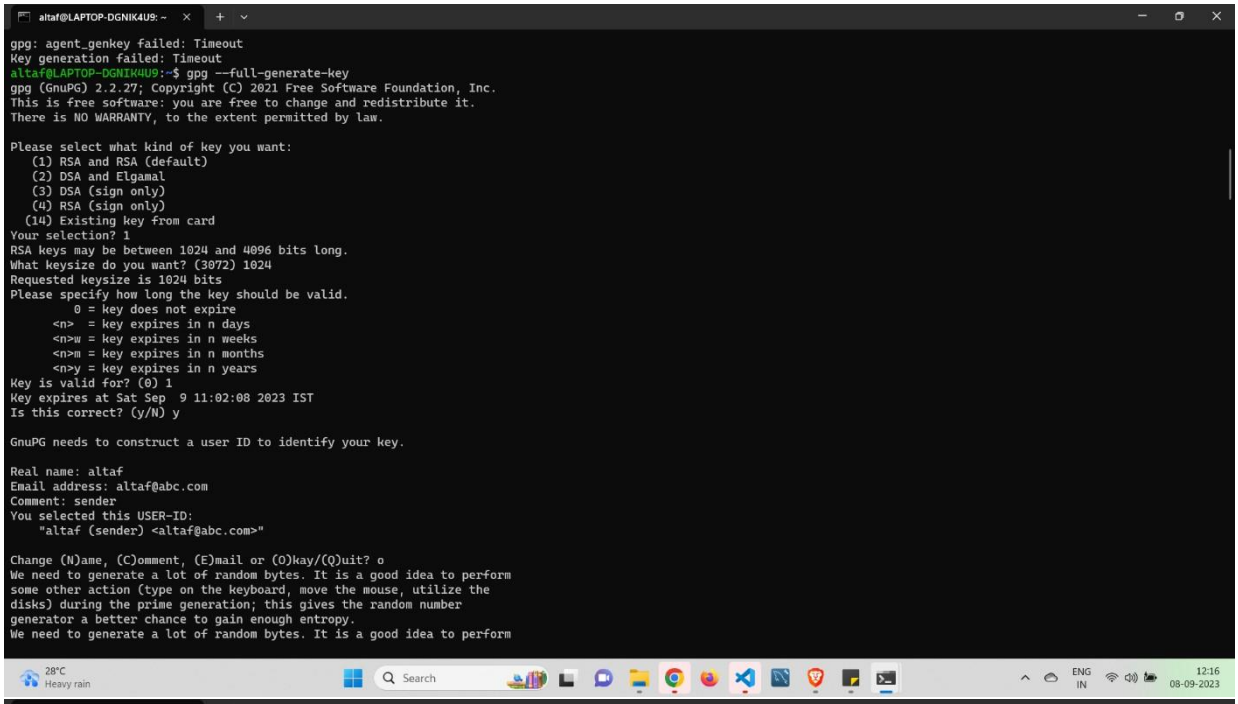
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23

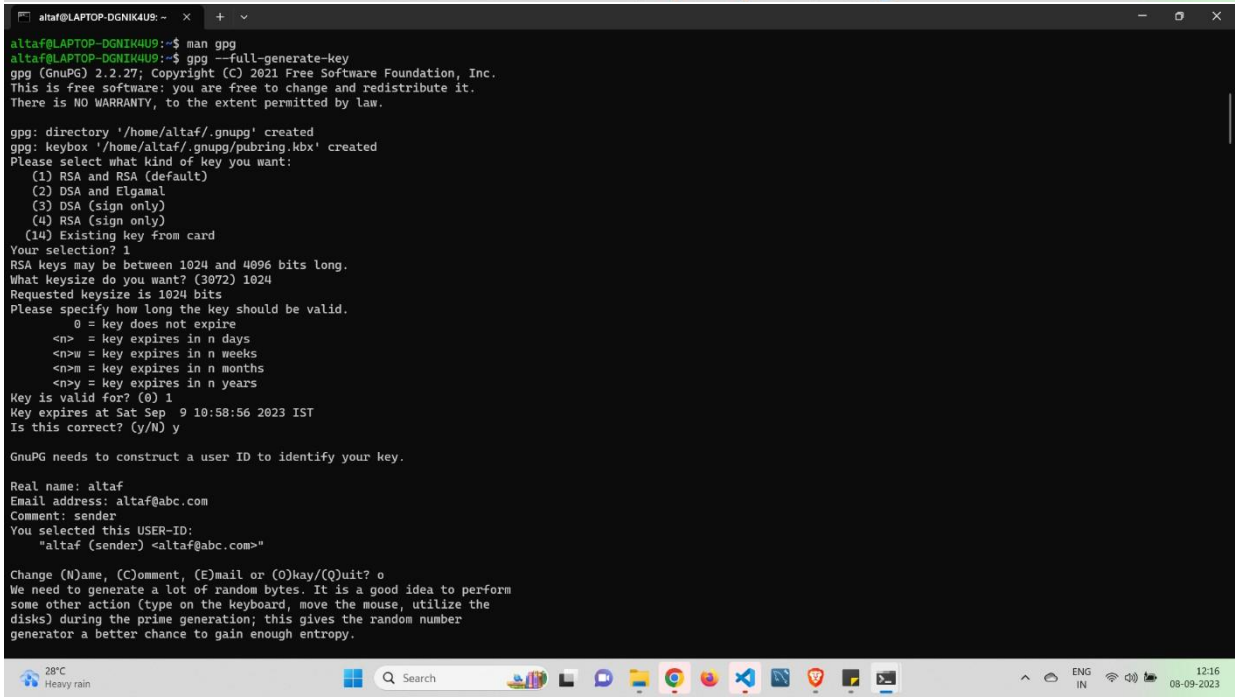




Name : Altaf Alam

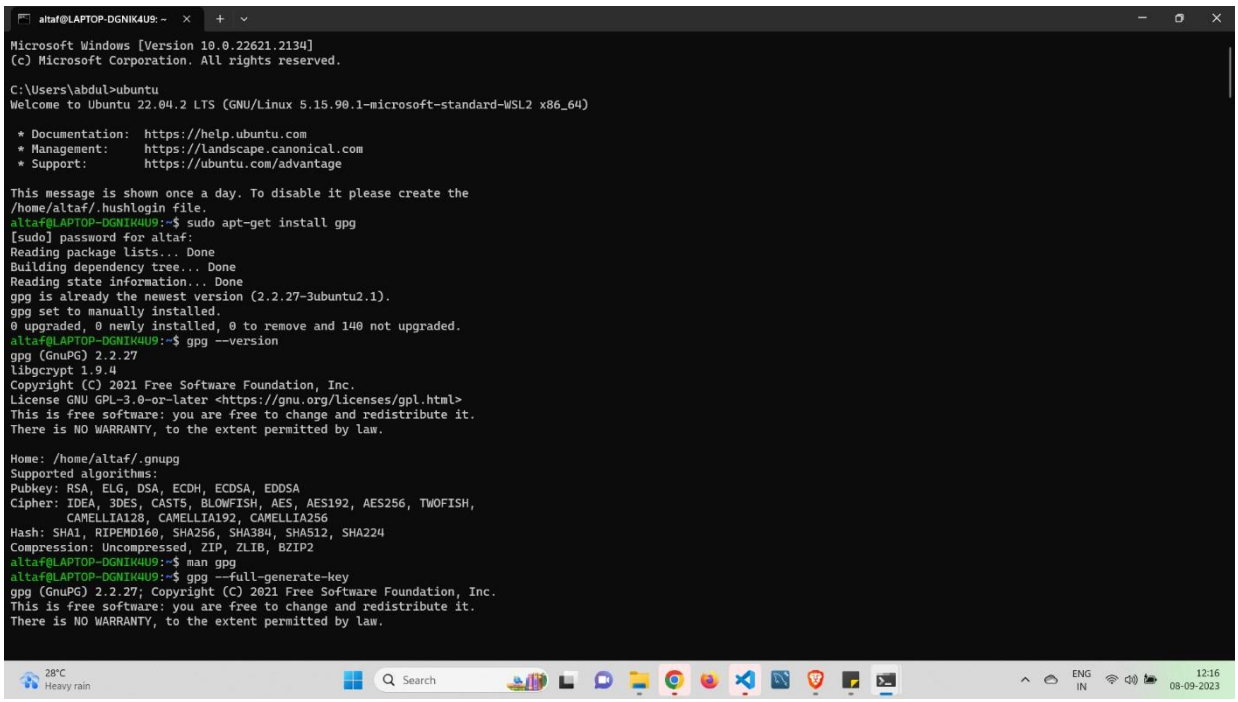
Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23





Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 14/09/23



Conclusion:

In summary, we explored GPG's private and public key rings, key management, and security processes. These are vital for secure communication and trust verification in digital exchanges.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 20/09/23

Assignment No 1

Padding Scheme Used in RSA Encryption: Why It Is Used and Its Limitations

RSA (Rivest-Shamir-Adleman) is one of the most widely used public-key encryption algorithms. It is used for securing communications, data transmission, and digital signatures. A crucial aspect of RSA encryption is the padding scheme, which is employed to address certain vulnerabilities and limitations of the basic RSA algorithm. In this detailed explanation, we will explore the padding scheme used in RSA, why it is necessary, and its limitations.

I. Introduction to RSA Encryption:

RSA encryption is a form of asymmetric cryptography, meaning it uses two distinct keys: a public key for encryption and a private key for decryption. The security of RSA relies on the mathematical difficulty of factoring large composite numbers into their prime factors. The fundamental equation in RSA is:

$$C \equiv M^e \mod N$$

Where:
- C is the ciphertext.
- M is the plaintext message.
- e is the public exponent. - N is the modulus (product of two large prime numbers).

The recipient, who possesses the private key with exponent d, can decrypt the ciphertext using the equation:

$$M \equiv C^d \mod N$$

While the basic RSA algorithm is secure when implemented correctly, it has certain vulnerabilities and limitations that necessitate the use of a padding scheme.

II. Why Padding is Used in RSA Encryption:

Padding in RSA encryption serves several essential purposes:

1. Preventing Attacks on Small Messages: Without padding, encrypting small messages directly using RSA can be insecure. An attacker could potentially guess the plaintext message by encrypting all possible messages and comparing the results to the ciphertext.

2. Adding Randomness: Padding introduces randomness into the encryption process. This randomness helps in ensuring that similar plaintexts do not produce the same ciphertext, which is essential for security.

3. Maintaining Consistent Block Size: RSA operates on blocks of data, and the size of these blocks must be consistent. Padding ensures that the plaintext message can be divided into these fixed-size blocks, even if it is not an exact multiple of the block size.

4. Meeting Cryptographic Standards: Various cryptographic standards, such as PKCS#1 (Public-Key Cryptography Standards), define specific padding schemes for RSA. Adhering to these standards ensures interoperability between different implementations of RSA encryption.

III. Types of Padding Schemes in RSA Encryption:

There are two primary padding schemes used in RSA encryption: PKCS#1 v1.5 padding (RSA-PKCS1) and OAEP (Optimal Asymmetric Encryption Padding). Let's briefly explain both:

1. PKCS#1 v1.5 Padding (RSA-PKCS1):
- PKCS#1 v1.5 padding is the older and more widely used padding scheme.
- It involves adding specific bytes to the plaintext message before encryption.
- These bytes include a block type identifier and random padding bytes.
- RSA-PKCS1 padding provides security against certain attacks, such as the Bleichenbacher attack, when correctly implemented. - However, it is vulnerable to attacks if not implemented carefully.

2. OAEP (Optimal Asymmetric Encryption Padding):
- OAEP is a more secure and modern padding scheme for RSA.
- It provides stronger security guarantees compared to PKCS#1 v1.5 padding.
- OAEP incorporates a cryptographic hash function, such as SHA-1 or SHA-256, to add randomness and additional security to the plaintext.
- It also includes error-detection capabilities, making it more robust. - OAEP is resistant to various attacks, including chosen-ciphertext attacks.

IV. Limitations of Padding in RSA Encryption:

While padding is essential for enhancing the security of RSA encryption, it is not without its limitations and challenges:

1. Padding Overhead: Padding increases the size of the plaintext message, leading to overhead in the encrypted data. In some cases, this overhead can be significant, especially for short messages.

2. Padding Vulnerabilities: As mentioned earlier, PKCS#1 v1.5 padding can be vulnerable to certain attacks, such as the Bleichenbacher attack, if not implemented correctly. Implementing padding schemes securely requires careful attention to detail.

3. Padding Schemes May Become Outdated: Over time, cryptographic standards and best practices evolve. Padding schemes that were once considered secure may become vulnerable to new attacks. It is essential to stay up-to-date with the latest cryptographic recommendations.

4. Performance Impact: The additional steps involved in padding, such as cryptographic hash functions, can introduce computational overhead. This can impact the performance of encryption and decryption operations, especially when dealing with large volumes of data.

5. Complexity: Implementing padding correctly can be complex. Errors in the implementation can lead to security vulnerabilities. Cryptographers and software developers must carefully follow standards and best practices.

V. Conclusion:

In summary, the padding scheme used in RSA encryption is a critical component for addressing vulnerabilities and limitations inherent in the basic RSA algorithm. Padding adds randomness, prevents certain attacks, and ensures consistent block sizes. While PKCS#1 v1.5 padding has been widely used, OAEP offers stronger security guarantees. However, both padding schemes come with their own trade-offs and must be implemented correctly to maintain the security of RSA encryption. Cryptographers and software developers must continually monitor developments in cryptography to adapt to evolving security threats and best practices in padding schemes and RSA encryption.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Security Lab , Date : 23/09/23

Assignment No 2

Introduction to Intrusion Detection Systems (IDS)

In the ever-evolving landscape of cybersecurity, the ability to detect and respond to security threats is paramount. Intrusion Detection Systems (IDS) are critical components of a robust cybersecurity strategy. They serve as vigilant sentinels, continuously monitoring network traffic and system behavior, ready to raise the alarm when unauthorized or malicious activities are detected. In this comprehensive exploration of IDS, we will delve into the various types of IDS, their working principles, and the advantages and limitations inherent in each type.

Types of Intrusion Detection Systems

1. Network-based IDS (NIDS):

Working Principle:
Network-based IDS, or NIDS, is designed to monitor and analyze network traffic. It operates at the network level, examining data packets as they traverse network segments or pass through designated inspection points. NIDS systems are particularly effective at identifying patterns or signatures that match known attack patterns. When a match is detected, an alert is generated.

Advantages:
- Effective for External Threats: NIDS is well-suited for detecting external threats, such as port scans, malware propagation, and network-based attacks.
- Visibility into Network Traffic: It provides a comprehensive view of network traffic, helping to identify suspicious activities across the entire network.

Limitations:
- Inability to Handle Encryption: NIDS struggles to inspect encrypted traffic, as it cannot decipher the contents of encrypted data packets.
- Evasion Techniques: It is vulnerable to evasion techniques used by attackers to disguise or obfuscate malicious traffic.
- Limited for Insider Threats: NIDS is less effective at detecting insider threats or attacks originating from within the network since it primarily monitors external traffic.

2. Host-based IDS (HIDS):

Working Principle:

Host-based IDS, or HIDS, operates at the individual host or endpoint level. It monitors activities on a specific system, examining system logs, configuration files, and system calls for signs of suspicious behavior. HIDS systems establish a baseline of normal host behavior and generate alerts when deviations from this baseline occur.

Advantages:

- Effective for Insider Threats: HIDS excels at detecting insider threats and attacks targeting specific hosts or endpoints.
- Detailed Host-level Information: It provides detailed information about host-level activities, making it easier to pinpoint the source of security incidents.

Limitations:

- Host-specific Monitoring: HIDS is confined to monitoring the host it is installed on and may not provide insights into broader network-level threats.
- Resource Intensive: Running HIDS on numerous hosts can be resource-intensive and impact the performance of the monitored systems.
- Less Effective for Network Attacks: HIDS is less effective at identifying network-level attacks, as it primarily focuses on host-level activities.

3. Anomaly-based IDS:

Working Principle:
Anomaly-based IDS takes a different approach by focusing on identifying deviations from established baselines of normal behavior. Instead of relying on predefined attack signatures, it uses statistical models, machine learning algorithms, or heuristics to detect anomalies in network traffic or system behavior. When an anomaly is detected, the system generates an alert.

Advantages:

- Detection of Unknown Attacks: Anomaly-based IDS can effectively detect previously unknown or zero-day attacks since it does not rely on predefined attack patterns.
- Adaptability: It is adaptable to evolving threats and changing network environments, as it can learn and adjust over time.

Limitations:

- High False Positives: During the initial training period, anomaly-based IDS often generates high numbers of false positives as it learns the baseline behavior.
- Difficulty in Discrimination: Distinguishing between legitimate anomalies and actual attacks can be challenging, leading to additional investigative work.

- Continuous Maintenance Required: Anomaly-based IDS systems require continuous updates and maintenance to ensure accurate anomaly detection.

4. Signature-based IDS (Misused-based IDS):

Working Principle:
Signature-based IDS, also known as misused-based IDS, relies on predefined signatures or patterns of known attacks. When network traffic or system behavior matches one of these signatures, an alert is generated. Signature-based IDS essentially compares incoming data packets or activities to a database of known attack patterns.

Advantages:
- High Accuracy for Known Attacks: Signature-based IDS systems exhibit high accuracy for detecting known attacks, making them reliable for identifying well-documented threats. - Low False Positives: They tend to have low false-positive rates, reducing the chances of generating alerts for benign activities.

Limitations:
- Ineffectiveness Against Novel Attacks: Signature-based IDS is ineffective against new or modified attacks that are not covered by existing signatures.
- Zero-day Vulnerability: It cannot detect zero-day attacks or novel vulnerabilities that have not yet been cataloged.
- Dependence on Signature Updates: Signature-based IDS systems require frequent updates to their signature databases to remain effective against evolving threats.

5. Behavior-based IDS (Heuristic-based IDS):

Working Principle:
Behavior-based IDS observes and analyzes the behavior of systems, applications, or users over time. It establishes a profile of normal behavior based on historical data and system behavior. When deviations from this baseline are detected, the system generates alerts.

Advantages:
- Detection of Unknown Threats: Behavior-based IDS is effective against previously unknown attacks and insider threats, as it focuses on behavioral anomalies.
- Contextual Analysis: It offers context for detected anomalies by considering multiple data points and the historical behavior of systems or users.

Limitations:

- Resource Intensive: Behavior-based IDS systems can be computationally intensive, requiring substantial resources for analysis.
- Potential for False Positives: If the baseline behavior is not accurately defined or is too broad, it can lead to false positives.
- Complex Configuration: Configuring and fine-tuning behavior-based IDS systems for specific environments can be complex and time-consuming.

Intrusion Detection Systems are indispensable components of modern cybersecurity strategies, serving as a critical line of defense against a wide range of threats. The choice of IDS type depends on an organization's specific security requirements, the nature of the threats it faces, and the available resources for maintenance and tuning. Many organizations opt for a combination of IDS types to create a comprehensive security posture, maximizing their ability to detect and respond to security incidents effectively.

Research on Computer Network Information Security System Based on Big Data

Gengyi Xiao
Department of Mathematics and Computer Technology, Guilin Normal College, Guilin, China
*Corresponding author e-mail: xiao6169@126.com

***Abstract*—In order to effectively improve the computer network security defence capability in the era of big data, a comprehensive analysis of the functions of big data centre applications is performed to create a comprehensive computing network security defence system. First of all, a comprehensive analysis of the hidden dangers of modern computer network security is carried out, and then corresponding technologies such as modern network security technology and solutions, intrusion detection technology are introduced to realize the design of computer network security defence system in the context of the big data era. After the system design is implemented, the system is tested accordingly. According to the test results, the computer network security defence system designed in this paper can actively discover and effectively prevent security threats in the network, thereby ensuring that the network can Normal and safe operation. The computing network security defence system can also provide effective ideas for future network security protection and achieve further expansion of security defence.**

Keywords—Big data era, network security, defence system, intrusion detection.

I. INTRODUCTION

As an iconic technology for human beings entering the 21st century, computer network technology has been continuously improved in China in the past decades. This has also provided a favourable background for the development of information technology. Different types of information methods are continuously integrated into people's daily life and production, bringing great convenience to people's lives, and higher industrial production efficiency. It can be said that the arrival of computer network technology has achieved social change. However, in the context of big data, computer network security has also received widespread attention, and information security issues are very serious. Personal information is transferred to big data, and anyone can query personal privacy information, which affects users. This requires the strengthening of computer network information security protection in the context of big data to protect user information interests.

II. OVERVIEW OF BIG DATA

A. Basic overview of big data

Since 2012, the term big data has been in people's field of vision and attracted constant attention. In the current period, Internet information technology is constantly expanding, and various information resources are flooding every corner of our lives, posing severe challenges for the development of enterprises in the future. Therefore, network information occupies a vitally important position in the operation and management of an enterprise [1]. In the context of the era of big data, information processing models have ushered in new changes, and are constantly being updated. Network information resources have been characterized by diversification and complexity. Therefore, major companies have contended for the market of network information resources, making it a unique advantage for their own development, and thus enhancing their competitiveness in development. It not only breaks the previous limitation of time and space, but also provides a broader information exchange platform for the operation and development of enterprises, and has become a treasure trove of resources that can be continuously tapped by enterprises in the development process. Therefore, constructing a perfect network information security system and continuously developing and using network information technology are important directions for the development of various enterprises in the current period.

B. Analysis of hidden dangers in big data security

The processing of big data includes processes such as generation, transmission, storage, analysis, push, and application. It involves data producers, software developers, distribution links, processors, and users. The information uploaded to the network is divided into structured data and There are two types of unstructured data. Structured data is stored in a relational database structure system. It has an obvious logical structure and is represented by a traditional two-dimensional table. In the era of big data, open data platforms are exposed to the eyes of professionals and various non-professionals. Anyone can send information to the server without strict review. Unstructured accounts for a larger proportion, including office documents in all formats. Comments, text, pictures, subsets of the standard universal mark-up language XML, HTML, various reports, images, audio, video, location information, and other media, the length of the field is variable, field duplication is legal, and You can set subfields and multivalued fields. The structuredness of big data makes the representation of data more difficult, and uniqueness and precision cannot be achieved. Traditional relational database management systems can only manage the structured part, but they are powerless and unstructured in the face of unstructured data. Although data management software has emerged, it has not yet reached a perfect level. There are loopholes in data protection, which provides hacking, information leakage, and Trojan horse penetration. It is difficult to trace the source according to system logs [2].

978-1-7281-7005-3/20/$31.00 ©2020 IEEE

C. Big data security risks

1) Privacy data leakage.

With the increase in the level of informatization, people's dependence on the Internet has become more serious. During online shopping, medical treatment, deposit and withdrawal, and social networking, the filled-in form contains a lot of private information, such as bank card account number, password, medical history, home address, ID card, and mobile phone number. Some are encrypted and some are stored in plain text. These data for merchants to analyse customer behaviour, targeting target groups and market predictions provide great convenience, but in the era of big data, the number of customers has become the main indicator of the potential value of the merchant. Driven by huge benefits, the phenomenon of buying and selling customer information is sometimes occurred. At the same time, due to technical reasons and non-standard prevention management, there is a possibility of hacking into the system, leading to the outflow of sensitive information.

2) Technical information leaked.

Enterprise technical documents, R & D data, and software source programs are stored in CRM, ERP, and OA systems. They are the core secrets of the enterprise. If they are not strictly controlled and intercepted by competitors or hackers through VPN, it will inevitably bring economic losses and because Vicious competition affects the production and operation of enterprises.

3) Government industry data outflow.

Household registration files, social security, provident funds, savings, personnel and other information are the guarantee of national security. If obtained by illegal invaders, it will not only pose a threat to the residents, but also cause social instability. In peacetime, government data and information agencies are often targeted by terrorist groups and extremists.

4) Internal data leakage.

Any website, computer information system and database have an administrator role, which can not only perform system management, but also directly enter the background to modify data. Driven by interests or personal purposes, they use administrator permissions to copy or illegally tamper with internal data, and very concealed and difficult to find [3].

III. The Importance of Computer Network Information Security

Due to the characteristics of openness, interconnection, diversity of connection methods, and uneven distribution of terminals, computer networks are vulnerable to computer viruses, hackers, or malicious software. In the face of various threats to network security, it is necessary to consider the crucial issue of network security. Taking enterprise information as an example, as enterprises pay more and more attention to the application of information data, they will inevitably establish websites or information platforms to collect and integrate information related to enterprise production. It directly induces security risks and brings certain negative effects to the enterprise. Therefore, attaching importance to computer network information security, actively finding potential threats in computer network information security, and formulating countermeasures are the key to the development and application of computer network technology in the context of current big data.

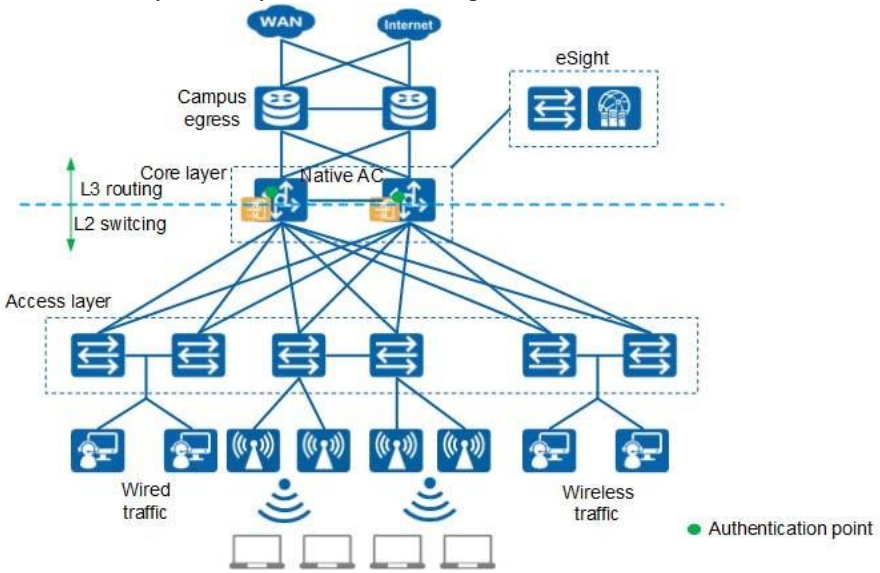
IV. Network Security Défense System Design

A. System requirements

The article uses a school as an example to implement the design of a computer network security defence system. The school campus network mainly includes 4 security levels, of which the first level of security requirements mainly includes the requirement for Internet access security; the system has the ability to restore; the identity authentication system is implemented. Design; the second-level security requirements include the interconnection of different subnetworks; the third-level security requirements are mainly to achieve secure

Figure 1. Overall campus network planning

The boundary firewall divides the network area and the includes the three subnets of the department office, campus network area. The network area mainly includes administrative office, and student computer room. Through subnets for external services. The campus network mainly the analysis of modern computer information network



access to services and intrusion detection. Figure 1 shows the network plan of the campus network [4]. system investigation and campus information system security requirements analysis, it can be said that the computer network security defence system requirements are mainly: desktop system security requirements, virus protection requirements, identity requirements, access control requirements, encryption requirements, security audits Requirements, intrusion security detection system requirements, vulnerability scanning requirements, security management requirements, and physical security measures.

B. Design of Network Security System

1) Security Défense Function.

2) Security protection.

At present, most computer network security defence systems use firewalls, antivirus software, and other content to achieve security protection. These software's are deployed individually or integrated, thereby effectively improving the integrity of big data application centres. In the process of the continuous promotion and popularization of modern big data

C. System test

Through the system designed in this paper, and the deployment of linkage devices is scanned using vulnerabilities, the school network security and the school intranet security can be effectively guaranteed. In the process of implementing department planning, it is necessary to be rational, so as to achieve the uniformity of department configuration strategies, to provide a basis for the configuration of subordinate departments and network management application service systems, and to ensure the network security of all departments. Figure 4 shows the deployment plan of the system [6].

The attack threats in the use of big data can be spread using multiple channels such as computers and mobile terminals. The latency period of Trojans and viruses is relatively long, which expands the scope of hacker's damage. In order to effectively improve the defence capabilities of big data application centres, you can create an active defence system, thereby further improving the ability of network security operations. Figure 2 shows the security defence system of a big data application centre.

application centres, the security defence measures also use digital signature defence technology to avoid repudiation in data communication. Therefore, the system designed in this article combines multiple defence technologies to prevent network data from being infected and attacked. Figure 3 shows the design of the security protection module [5].

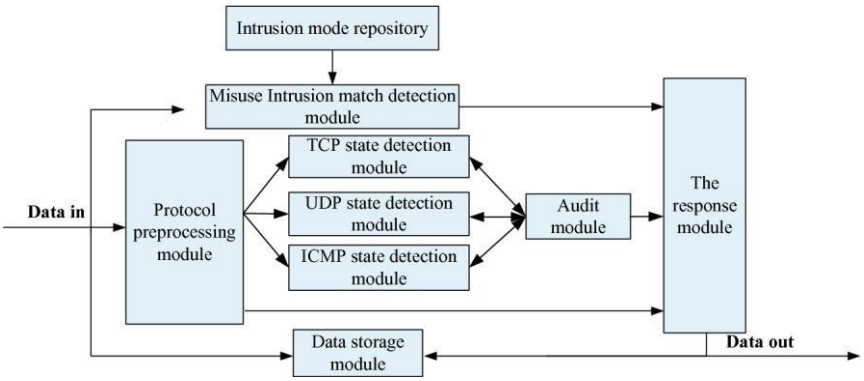


Figure 3. Security module design

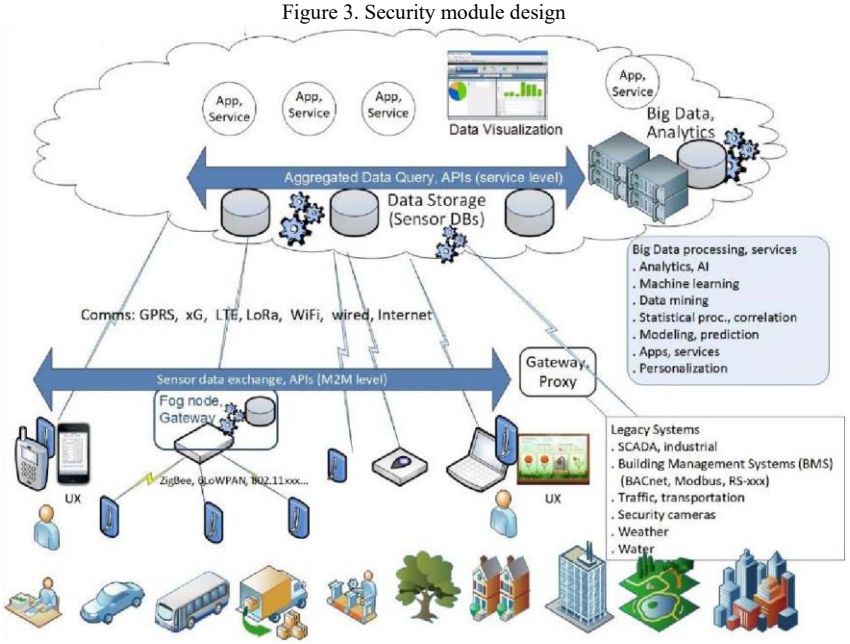


Figure 2. Security Défense System of Big Data Application Centre

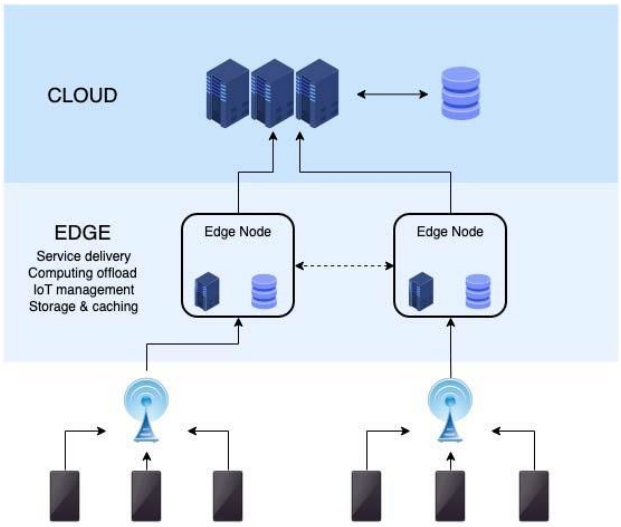


Figure 4. System test deployment scheme

V. Measures to Prevent Computer Network Information Security Risks in the Context of Big Data

A. Strengthening Account Password Security

When using a computer network system, various accounts such as computer system accounts, online banking accounts, and email accounts are inevitably involved. These accounts involve the privacy of the user. Once the account password is leaked, it will inevitably cause some damage to normal life. Impact. Specific methods to strengthen account password security include: setting difficult and complicated passwords, using numbers and letters as much as possible to reduce the possibility of password cracking; avoiding the use of numbers from ID cards or other documents as passwords; avoiding the use of multiple websites The same set of user names and passwords, especially accounts involving personal property such as online banking; to avoid leaking personal information, try not to visit illegal websites.

B. Improve the environment for system operation

For some important hardware equipment, maintenance and treatment should be carried out regularly. If problems are found, they should be handled in time to ensure the normal use of the entire hardware equipment. In addition, relevant technical staff should also be provided with special technical training. The training content includes data security prevention knowledge and the characteristics of the Internet to improve their comprehensive professional quality. In addition, the registration management system should be improved. For example, for the maintenance of some equipment, registration work should be done to ensure that the system configuration, technical parameters, management and maintenance, and data performance of the social security system are in the best state.

C. Check the security performance of the terminal in time

Basic-level terminal equipment is the target of hacking in the actual use process, and is easily affected by viruses and Trojan horse programs. In this regard, the security of the access terminal needs to be strictly checked, especially the update and installation of virus software on the terminal, and the operating system patches must be updated in time to eliminate potential security risks in the terminal.

D. Implementing Intrusion Detection

Intrusion detection is a network prevention method using network communication technology, artificial intelligence technology and other monitoring methods. It is divided into two methods: statistical analysis method and signature analysis method. The statistical analysis method is based on statistical principles, and comprehensively analyses the system's action mode under normal conditions to determine whether there is an abnormality in the realtime action of the system. The signature analysis method is based on the known vulnerabilities of the system to prevent security. Through template matching, the existing attack mode A problem was found in the signature. Intrusion detection can effectively detect computer network information security risks, and certain measures can be taken in time to avoid the risks.

VI. Conclusion

During the continuous development of big data technology, the channels of network attacks are constantly changing, and the latency period of network attacks is not only increasing, but the speed of security threat infection is also getting faster and faster. Influence. Therefore, it is necessary to regularly use advanced security countermeasures to effectively improve the security defence capabilities of big data application centres, thereby achieving in-depth defence of network threats.

Acknowledgments

This work was financially supported by Guangxi university scientific research fund: Research on the key technologies of vehicle-mounted network based on CPS (2013YB286).

References

[1] S. Vijayakumar Bharathi. Prioritizing and ranking the big data information security risk spectrum. Global Journal of Flexible Systems Management, 18(2) (2017) 183-201.

[2] James T. Graves, Alessandro Acquits, & Nicolas Christin. Big data and bad data: on the sensitivity of security policy to imperfect information. University of Chicago Law Review, 83(1) (2016)117137.

[3] Santosh Aditham, & Nagarajan Ranganathan. A system architecture for the detection of insider attacks in big data systems. IEEE Transactions on Dependable and Secure Computing, 15(6) (2018) 974-987.

[4] Xiaoming Wang, Carolyn Williams, Zhen Hua Liu, & Joe Croghan. Big data management challenges in health research. Briefings in Bioinformatics, 20(1) (2017) 1-12.

[5] Wu, Z., Niu, F., Pan, D., & Lei, J. Authority for swim based on attribute encryption., 43(3) (2017) 350-357.

[6] Liu, Y., Wang, X., Zhang, J., Zhang, M., Peng, L., & Xu, A. W. An improved security 3d watermarking method using computational integral imaging cryptosystem., 12(2) (2016) 1-21.

[7] Sharma Kartik; Aggarwal Ashutosh; Singhania Tanay; Gupta Deepak; Khanna Ashish (2019). Hiding Data in Images Using Cryptography and Deep Neural Network. Journal of Artificial Intelligence and Systems, 1, 143–162.

[8] G. H. Rosa, J. P. Papa (2019). Soft-Tempering Deep Belief Networks Parameters Through Genetic Programming. Journal of Artificial Intelligence and Systems, 1, 43–59.

Research on Computer Network Information Security System Based on Big Data

Altaf Alam - 02
Prasad Arote - 06
Krish Chaurasiya - 12
Yash Dave - 19

Introduction to Computer Network Technology

1. Improvement in computer network technology.
2. Favors the development of information technology.
3. Enhances convenience and industrial production efficiency.
4. Achieves social change .
5. Emphasize the Importance of Computer network security.

Basic Overview of Big Data

1. Emergence of the term "Big Data" since 2012.
2. Rapid expansion of internet information technology.
3. Proliferation of information resources.
4. Challenges for future enterprise development.
5. Vital role of network information in enterprise operations.

Hidden Dangers Big Data Security

1. Big Data includes
   - Structured Data
   - Unstructured Data

2. Traditional Relational Databases cannot be used for unstructured data.

3. Data Management Software are emerging but they have some issue in data protection.

4. Thus the openness and complexity of Big Data can create security risks.

Big Data Security Risks

1.Privacy Data Leakage:

- Data used for analysis and target marketing.
- Increased Security Risks such as data breaches, hacking

2. Technical Information Leakage:

- Sensitive(core data) of enterprise stored in critical systems(ERP,CRM etc).
- Vulnerabilities via VPNs by competitors or hackers.

3. Government Data Outflow:

- Security of government databases is essential for national security.
- Illegal access to information poses threat to citizens and can lead to social instability.

4. Internal Data Leakage:

- Computer Information Systems have administrative roles with significant privileges.
- Driven by personal interests the administrators may copy or tamper the data.

Importance of Computer Network Information Security

A. Vulnerability of Computer Networks:

- Openness, interconnection, diverse connection methods, and uneven terminal distribution make computer networks susceptible to threats.
- Threats include computer viruses, hackers, and malicious software.

B. Impact on Enterprises:

- Enterprises emphasize information data application and often establish websites and information platforms for data integration.
- This integration introduces security risks and can have negative effects on businesses.

` C. Crucial Network Security:

- Given the growing importance of information data, network security becomes paramount.
- Actively identifying potential threats and formulating countermeasures are essential to protect businesses in the era of big data.

Measures to prevent Security Risks

A. Intrusion Detection:

- Network Defense: Intrusion detection utilizes network communication and AI.
- a) Statistical Analysis: Detect anomalies in system behavior.
- b) Signature Analysis: Identify known vulnerabilities.
- Effective Risk Mitigation: Detect and respond to network security threats promptly.

B. Terminal Security:

- Hacker Targets: Basic terminals are frequent hacker targets.
- Security Measures: Rigorously inspect access terminals, keep antivirus software updated, and maintain current OS patches to eliminate security risks.

Measures to prevent Security Risks

C. Enhancing System Operation:

- Regular Maintenance: Ensure hardware operates smoothly.
- Technical Training: Educate staff on data security and internet characteristics.
- Improved Registration: Implement equipment registration for optimal system setup, parameters, and performance.

D. Strengthening Account Password Security:

- Importance: Protect privacy across various accounts (e.g., systems, online banking, email).
- Password Tips: Create complex, alphanumeric passwords, Avoid common choices (e.g., ID card numbers), Use unique login credentials for each account, Refrain from visiting unsafe websites.

NETWORK SECURITY DÉFENSE SYSTEM DESIGN

1. System Requirements:

- Level 1 focuses on internet access security, data restoration, and user identity verification.
- Level 2 addresses the interconnection of different subnetworks.
- Level 3 ensures secure access to services and intrusion detection.

2. Security Defense Function:

- An active defense system is proposed to enhance network security and proactively safeguard against a range of threats, including viruses and Trojans, known for their potential to cause substantial damage with prolonged latency periods.
- This system is designed to actively improve network security operations by countering diverse threats and attacks.

NETWORK SECURITY DÉFENSE SYSTEM DESIGN

3. Security Protection:

- Common security measures like firewalls, antivirus software, and digital signatures work together to strengthen the integrity of the network.
- The system utilizes a blend of defense technologies to protect against data infection and attacks, ensuring the security of data during communication.

4. System Testing:

- Employing vulnerability scans to identify and address network weaknesses.
- The deployment plan plays a pivotal role in securing the entire school network, ensuring consistent configurations, and facilitating effective network management.

References

Reference to IEEE paper- https://drive.google.com/file/d/1f2XOniVxYm6afWEDXAqt0HkhprrVfCzo/view?usp=drive_link