Basics of Information Security (4360702)

Laboratory Manual

Semester-6

Diploma in Computer Engineering

Enrolment Number	
Name	
Branch	Computer Engineering
Academic Term	2025-26 (Even Term)
Institute	AVPTI, Rajkot

Directorate Of Technical Education

Gandhinagar - Gujarat

DTE's Vision:

To facilitate quality technical and professional education having relevance for both industry and society, with moral and ethical values, giving equal opportunity and access, aiming to prepare globally competent technocrats.

DTE's Mission:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

Institute's Vision:

To cater skilled engineers having potential to convert global challenges into opportunities through embedded values and quality technical education.

Institute's Mission:

- Impart quality technical education and prepare diploma engineering professionals to meet the need of industries and society.
- Adopt latest tools and technologies for promoting systematic problem solving
  skills to promote innovation and entrepreneurship
- Emphasize individual development of students by inculcating moral, ethical and life skills.

Department's Vision:

Develop globally competent Computer Engineering Professionals to achieve excellence in an environment conducive for technical knowledge, skills, moral values and ethical values with a focus to serve the society

Department's Mission:

- To provide state of the art infrastructure and facilities for imparting quality education and computer engineering skills for societal benefit.
- Adopt industry-oriented curriculum with an exposure to technologies for building systems & application in computer engineering
- To provide quality technical professional as per the industry and societal needs, encourage entrepreneurship, nurture innovation and life skills in consonance with latest interdisciplinary trends.

Certificate

This is to certify that Mr./Ms. ……. Enrolment No. ……. of Semester: 6th of Diploma in *Computer Engineering* of Institute AVPTI, Rajkot (GTU Code: 602) has satisfactorily completed the term work in course *Basic of Information Security (4360702)* for the Academic Year: 2025-26 (Even Term) as prescribed in the GTU curriculum.

Place: Rajkot

Date: ……

Faculty Signature

Programme Outcomes (POs):

Following programme outcomes are expected to be achieved through the practical of the course:

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyse well-defined engineering problems using codified standard methods.
3. **Design/development of solutions:** Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
7. **Life-long learning:** Ability to analyse individual needs and engage in updating in the context of technological changes in field of engineering.

Program Specific Outcomes (PSOs)

- Able to apply the knowledge gained from Mathematics, Basic Sciences in general and all computer science courses in particular to identify, formulate and solve real life complex engineering problems faced in industries and society.
- The ability to employ modern computer languages, environments and platforms in creating innovative career paths in Hardware, Networking and Software Development technologies.

Practical Outcome - Course Outcome Matrix

Course Outcomes (COs):

1) Describe fundamentals of information security.
2) Demonstrate substitution, transposition technique and symmetric cryptography algorithm.
3) Demonstrate the public key encryption with public key cryptography.
4) Apply measures to protect the network communication from attacks using firewalls and intrusion detection systems.
5) Describe the basics of cyber security, cyber-attacks, cybercrime.

Sr.	Practical Outcome	CO1	CO2	CO3	CO4	CO5
1.	Execute Basic TCP/IP utilities and commands.	√	-	-	-	-
2.	Write a Program to implement Caesar Cipher for basic encryption and decryption. (Using Any of the Language C/C++/Java/Python)	-	√	-	-	-
3.	Write a Program to implement Hill Cipher for basic encryption techniques. (Using Any of the Language C/C++/Java/Python)	-	√	-	-	-
4.	Write a Program to implement the Play-Fair Cipher Technique for encryption. (Using Any of the Language C/C++/Java/Python)	-	√	-	-	-
5.	Write a Program to implement the Rail Fence Technique for encryption. (Using Any of the Language C/C++/Java/Python)	-	√	-	-	-
6.	Write a Program to implement RSA algorithm for asymmetric key encryption. (Using Any of the Language C/C++/Java/Python)	-	-	√	-	-
7.	Simulate the concept of Virtual LAN using Cisco Packet Tracer.	-	-	-	√	-
8.	Simulate the working of Firewall using Cisco Packet Tracer.	-	-	-	√	-
9.	Study cyber security fundamentals, including common threats and mitigation strategies.	-	-	-	-	√
10.	Study of Kali Linux Operating System for Cybersecurity.	-	-	-	-	√

Progressive Assessment Sheet / Index

Sr.	Experiment Name/Practical Outcome	CO	Date	Marks (25)	Sign
1	Execute Basic TCP/IP utilities and commands.	CO1			
2	Write a Program to implement Caesar Cipher for basic encryption and decryption. (Using Any of the Language C/C++/Java/Python)	CO2			
3	Write a Program to implement Hill Cipher for basic encryption techniques. (Using Any of the Language C/C++/Java/Python)	CO2			
4	Write a Program to implement the Play-Fair Cipher Technique for encryption. (Using Any of the Language C/C++/Java/Python)	CO2			
5	Write a Program to implement the Rail Fence Technique for encryption. (Using Any of the Language C/C++/Java/Python)	CO2			
6	Write a Program to implement RSA algorithm for asymmetric key encryption. (Using Any of the Language C/C++/Java/Python)	CO3			
7	Simulate the concept of Virtual LAN using Cisco Packet Tracer.	CO4			
8	Simulate the working of Firewall using Cisco Packet Tracer.	CO4			
9	Study cyber security fundamentals, including common threats and mitigation strategies.	CO5			
10	Study of Kali Linux Operating System for Cybersecurity.	CO5			

Rubrics for Continuous Assessment- CA (25 Marks)

Component	Criteria	Marks	Assessment
Laboratory Work and Questionnaire	**Excellent**	**(23-25)**	Demonstrates exceptional proficiency in both laboratory work and questionnaire assessments, consistently applying skills and understanding effectively.
	Proficient	**(18-22)**	Shows a strong command of both laboratory work and questionnaire assessments, with minor areas for improvement.
	Satisfactory	**(13-17)**	Achieves a satisfactory level of performance in laboratory work and questionnaire assessments, with room for improvement in some areas.
	Needs Improvement	**(8-12)**	Demonstrates limited proficiency in both laboratory work and questionnaire assessments, with significant areas for improvement.
	Inadequate	**(0-7)**	Fails to meet acceptable standards in both laboratory work and questionnaire assessments; significant improvement is required.

Practical-1 : Execute Basic TCP/IP utilities and commands.

Objective:

To utilize basic TCP/IP utilities and commands to troubleshoot network issues, gather configuration information, and analyse connectivity, aiding in effective network management and diagnostics.

Expected Program Outcomes (POs): PO1, PO5 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- **Network Troubleshooting**: Identify and fix connectivity issues using tools like ping and arp.
- **IP Configuratio**n: View and manage IP settings with ipconfig (Windows) or ifconfig (Linux).
- **Routing Analysis**: Trace packet routes using tracert (Windows) or traceroute (Linux).
- **ARP Management**: Understand and troubleshoot ARP for address resolution.
- **Traffic Analysis**: Capture and analyse network traffic with tcpdump.
- **Domain Information**: Retrieve domain and IP details using whois, host, and nslookup.
- **Connection Monitoring**: Monitor network connections and ports using netstat.
- **File Transfer**: Transfer files using FTP and understand basic telnet for remote connections.

Expected Course Outcomes (Cos): CO1

Practical Outcome (PRo): Execute Basic TCP/IP utilities and commands.

Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

Prerequisite Theory:

TCP/IP Model:

Explanation: The TCP/IP model is a conceptual framework that standardizes the functions of a telecommunication or computing system into different layers. It consists of the Application, Transport, Network, Data Link, and Physical layers.

Example: The process of sending an email involves multiple layers of the TCP/IP model, from the application layer (email client) to the physical layer (network hardware).

IP Addressing:

Explanation: IP addressing is a system used to uniquely identify devices on a network. IPv4 addresses are written as four sets of numbers, and IPv6 uses a longer hexadecimal format.

Example: IPv4 address - 192.168.1.1, IPv6 address - 3001:0db8:85a3:0000:0000:8a2e:0370:7334

DNS (Domain Name System):

Explanation: DNS translates human-readable domain names into IP addresses. It consists of a hierarchical system of DNS servers.

Example: Resolving the domain "www.example.com" to the IP address 203.0.113.5 using DNS.

Routing:

Explanation: Routing is the process of forwarding data packets between different networks. Routers use routing tables to determine the best path for packet delivery.

Example: A router deciding the best path for a packet to travel from one network to another based on the destination IP address.

Network Protocols:

Explanation: Network protocols define rules and conventions for communication between devices on a network.

Example: TCP ensures reliable, connection-oriented communication, while UDP provides faster but less reliable communication for activities like streaming.

Ethernet and MAC Addresses:

Explanation: Ethernet is a widely used LAN technology, and MAC addresses are unique identifiers assigned to network interfaces.

Example: A MAC address like "00:1A:2B:3C:4D:5E" uniquely identifies a network device.

Firewall Concepts:

Explanation: Firewalls are network security devices that control incoming and outgoing network traffic based on an organization's predefined security rules.

Example: Configuring a firewall to allow or block specific ports for incoming and outgoing traffic.

FTP and Telnet Protocols:

Explanation: FTP is a protocol for transferring files between computers, while Telnet provides a text-based interface for remote access.

Example: Using FTP to upload/download files to/from a server or using Telnet to remotely access and manage a router.

Network Security Basics:

Explanation: Network security involves protecting data integrity, confidentiality, and availability. It includes measures like encryption, access controls, and regular security audits.

Example: Implementing WPA2/WPA3 encryption for Wi-Fi networks to secure wireless communication.

Command-Line Interface (CLI):

Explanation: CLI is a text-based interface where users interact with a computer by typing commands. It is efficient for network configuration and troubleshooting.

Example: Using the Command Prompt (Windows) or Terminal (Linux) to execute commands like ping or ipconfig/ifconfig.

Network Topology:

Explanation: Network topology defines the physical or logical layout of interconnected devices in a network.

Example: A star topology where all devices are connected to a central hub or a bus topology where devices share a common communication line.

Network Commands

Ping: ping www.example.com

This command sends ICMP echo requests to the specified domain (www.example.com in this case) and measures the response time.

Ipconfig (Windows) / ifconfig (Linux):

ipconfig /all (Windows) ifconfig (Linux)

These commands display detailed information about your network interfaces, IP addresses, and related configuration settings.

Tracert (Windows) / traceroute (Linux):

tracert www.example.com (Windows) traceroute www.example.com (Linux)

These commands trace the route that packets take to reach a destination, showing the IP addresses of each hop.

ARP:

arp -a (Windows)

This command displays the ARP (Address Resolution Protocol) cache, showing the mappings between IP addresses and MAC addresses.

Tcpdump:

tcpdump -i eth0

This command captures and displays packets on a specific network interface (eth0 in this case) in real-time.

Whois:

whois example.com

This command provides information about the domain registration, including registrar details and registration dates.

Host:

host www.example.com

This command translates a domain name to its corresponding IP address.

Netstat:

netstat -an

This command displays active network connections, listening ports, and related information.

Nslookup:

nslookup www.example.com

This command performs DNS (Domain Name System) lookups, providing information about the specified domain.

FTP:

ftp ftp.example.com

This command opens an FTP (File Transfer Protocol) session to the specified FTP server, allowing you to transfer files.

Telnet:

telnet www.example.com 80

This command opens a telnet session to the specified host and port, allowing you to interact with a service (in this case, port 80 is often used for HTTP).

Task Do be Done: List all the basic TCP/IP commands with their use.

Practical No-2: Write a Program to implement Caesar Cipher for basic encryption and decryption.

Objective:

Develop a Python program to implement the Caesar Cipher for encrypting and decrypting text, showcasing fundamental cryptographic principles.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- **Algorithmic Implementation**: Develop the ability to implement the Caesar Cipher algorithm effectively, understanding the steps involved in both encryption and decryption.
- **Coding Proficiency in Python**: Enhance Python programming skills by writing code for user input, string manipulation, and modular arithmetic required for the Caesar Cipher.
- **Cryptography Awareness**: Gain an understanding of basic cryptographic concepts, especially in the context of symmetric key encryption, as demonstrated by the Caesar Cipher.
- **Debugging and Troubleshooting**: Cultivate skills in identifying and fixing errors in the code, ensuring the program works correctly for various input scenarios.
- **User Interaction and Input Handling**: Learn to interact with users by handling input for the text to be encrypted or decrypted, providing a practical application of user input processing in programming.

Expected Course Outcomes (Cos): CO2

Practical Outcome (PRo): Write a Program to implement Caesar Cipher for basic encryption and decryption.

Expected Affective domain Outcome (ADos)

- Enhanced Confidence in Programming.
- Developed Problem-Solving Skills.
- Appreciation for Cryptographic Concepts.
- Critical Thinking in Security Context

Prerequisite Theory:

Symmetric Key Encryption:

Familiarity with the concept of symmetric key cryptography, where the same key is used for both encryption and decryption.

Basic Python Programming:

Knowledge of basic Python syntax, including variables, data types, input/output, and string manipulation.

Modular Arithmetic:

Understanding modular arithmetic concepts, specifically modulo operations, which are fundamental to the Caesar Cipher algorithm.

Cryptography Basics:

Awareness of basic cryptographic terms such as plaintext, ciphertext, encryption, and decryption.

Caesar Cipher Algorithm:

Knowledge of how the Caesar Cipher works, including the shift operation and its impact on the encryption and decryption processes.

User Input Handling:

Understanding how to take user input in Python and handle it appropriately for further processing.

Algorithmic Thinking:

Basic algorithmic thinking skills, including the ability to break down a problem into smaller steps and devise a systematic solution.

Security Considerations:

An introductory understanding of security concepts, especially the limitations and vulnerabilities of simple encryption methods like the Caesar Cipher.

Definition and example of Caesar Cipher

The Caesar Cipher is a simple substitution cipher in cryptography, where each letter in the plaintext is shifted a certain number of positions down or up the alphabet. It is a type of symmetric key algorithm, meaning the same key is used for both encryption and decryption. The cipher is named after Julius Caesar, who is historically known to have used it to encrypt his private correspondence.

Here's a brief explanation of how the Caesar Cipher works with an example:

Algorithm:

Key Generation:

Choose a shift value, often referred to as the "key." This key determines the amount by which each letter will be shifted.

Encryption:

For each letter in the plaintext, shift it by the key value.

Wrap around to the beginning of the alphabet if the shift goes beyond 'Z' (for positive shifts) or to the end if it goes before 'A' (for negative shifts).

Maintain the case of the letters (uppercase or lowercase).

Decryption:

Decryption is essentially the reverse process of encryption.

For each letter in the ciphertext, shift it back by the key value.

Example:

Let's use a Caesar Cipher with a shift of 3:

Plaintext: "HELLO"

Encryption:

H -> K (shifted 3 positions)

E -> H

L -> O

L -> O

O -> R

Ciphertext: "KHOOR"

To decrypt "KHOOR" back to "HELLO," we perform the reverse shift:

Ciphertext: "KHOOR"

Decryption:

K -> H (shifted back 3 positions)

H -> E

O -> L

O -> L

R -> O

Plaintext: "HELLO"

Procedure to be followed/Source code with Output:

Practical No-3: Write a Program to implement Hill Cipher for basic encryption techniques.

Objective:

Develop a Python program to implement the Hill Cipher for encrypting and decrypting text, showcasing fundamental cryptographic principles.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- **Algorithmic Implementation**: Develop the ability to implement the Caesar Cipher algorithm effectively, understanding the steps involved in both encryption and decryption.
- **Coding Proficiency in Python**: Enhance Python programming skills by writing code for user input, string manipulation, and modular arithmetic required for the Caesar Cipher.
- **Cryptography Awareness**: Gain an understanding of basic cryptographic concepts, especially in the context of symmetric key encryption, as demonstrated by the Caesar Cipher.
- **Debugging and Troubleshooting**: Cultivate skills in identifying and fixing errors in the code, ensuring the program works correctly for various input scenarios.
- **User Interaction and Input Handling**: Learn to interact with users by handling input for the text to be encrypted or decrypted, providing a practical application of user input processing in programming.

Expected Course Outcomes (Cos): CO2

Practical Outcome (PRo): Write a Program to implement Hill Cipher for basic encryption and decryption.

Expected Affective domain Outcome (ADos)

- Enhanced Confidence in Programming.
- Developed Problem-Solving Skills.
- Appreciation for Cryptographic Concepts.
- Critical Thinking in Security Context.

Prerequisite Theory:

Matrix Operations: Familiarity with basic matrix operations such as addition, subtraction, multiplication, and inverse.

Linear Algebra Basics: Understanding of concepts like determinants, adjugated, and matrix inversion.

Modular Arithmetic: Similar to the Caesar Cipher, knowledge of modular arithmetic is crucial for certain calculations in the Hill Cipher.

Symmetric Key Cryptography: An understanding of symmetric key cryptography, as the Hill Cipher is a symmetric key algorithm.

Encryption and Decryption Principles: General knowledge of encryption and decryption principles in the context of cryptography.

Programming Fundamentals: Basic understanding of programming concepts, especially in the chosen programming language (e.g., Python).

ASCII Representation of Characters: Awareness of how characters are represented in the ASCII (American Standard Code for Information Interchange) system, as this is often used in programming.

Hill Cipher Algorithm: Understanding the Hill Cipher algorithm, including the process of key generation, encryption, and decryption.

Key Space in Cryptography: Familiarity with the concept of key space and its importance in the security of cryptographic algorithms.

Security Considerations: Awareness of potential vulnerabilities and security considerations related to the Hill Cipher.

Definition and example of Hill Cipher

The Hill Cipher is a symmetric key algorithm used for encryption and decryption. It operates on blocks of plaintext (usually pairs or triplets of letters), and its key is represented by a matrix. Here's an explanation of how the Hill Cipher works with a simple example:

Algorithm:

Key Generation:

Choose a key matrix, typically a square matrix of a certain size (e.g., 2x2, 3x3).

Ensure the matrix is invertible (its determinant is not zero) to enable decryption.

Matrix Representation of Plaintext:

Convert each pair or triplet of letters from the plaintext into numerical values using a mapping (e.g., A=0, B=1, ..., Z=25).

Encryption:

Represent the plaintext as a column matrix.

Multiply the key matrix with the plaintext matrix (modulo the alphabet size).

Convert the result back to letters using the reverse mapping.

Decryption:

Multiply the ciphertext matrix by the inverse of the key matrix (modulo the alphabet size).

Convert the result back to letters using the reverse mapping.

Examples:

Input: Plaintext: ACT

Key: GYBNQKURP

Output: Ciphertext: POH

Input: Plaintext: GFG

Key: HILLMAGIC

Output: Ciphertext: SWK

Encryption

We have to encrypt the message 'ACT' (n=3). The key is 'GYBNQKURP' which can be written as the nxn matrix:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

The message 'ACT' is written as vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The enciphered vector is given as:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

which corresponds to ciphertext of 'POH'

Decryption

To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters). The inverse of the matrix used in the previous example is

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} \equiv \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \pmod{26}$$

For the previous Ciphertext 'POH':

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \equiv \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} \pmod{26}$$

Procedure to be followed/Source code with Output:

Practical No- 4: Write a Program to implement the Play-Fair Cipher Technique for encryption.

Objective:

Develop a Python program to implement the Play-Fair Cipher for encrypting and decrypting text, showcasing fundamental cryptographic principles.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- **Programming Language Knowledge**: Choose a programming language you are comfortable with. Common choices include Python, Java, C++, or any language you are proficient in.
- **Understanding of Playfair Cipher**: Familiarize yourself with the Playfair Cipher algorithm. Understand how it works and the key steps involved in encryption.
- **Input Handling**: Implement a function to handle the input, which typically involves taking a key and the plaintext as input.
- **Key Square Generation**: Create a key square (5x5 matrix) based on the given key. The key square is used to encrypt and decrypt the message.
- **Text Preprocessing**: Preprocess the plaintext by removing spaces, converting to uppercase, and handling repeated letters.
- **Encryption Logic**: Implement the main logic for Playfair Cipher encryption. This involves dividing the plaintext into pairs, determining the positions of the letters in the key square, and applying the specific rules of the Playfair Cipher.

Expected Course Outcomes (Cos): CO2

Practical Outcome (PRo): Write a Program to implement Play-Fair Cipher for basic encryption and decryption.

Expected Affective domain Outcome (ADos)

- Interest and Engagement.
- Persistence.
- Collaboration and Communication.
- Critical Thinking in Security Context.
- Ethical Considerations

Prerequisite Theory:

Programming Fundamentals: Learners should have a good grasp of basic programming concepts such as variables, data types, control structures (if statements, loops), functions, and arrays or lists.

Data Structures: Understanding basic data structures is crucial. In the case of the Playfair Cipher, knowledge of matrices or 2D arrays is particularly important for implementing the key square.

String Manipulation: The ability to manipulate strings is essential for handling plaintext and processing it according to the Playfair Cipher rules. This includes operations like removing spaces, converting characters to uppercase, and breaking the text into pairs.

Functions and Modularization: Familiarity with functions and the concept of modular programming is important. Breaking down the encryption process into smaller, manageable functions can improve code organization and readability.

Algorithms and Logic: Understanding basic algorithms and logical reasoning is crucial for implementing the encryption logic of the Playfair Cipher. This includes identifying letter pairs, determining their positions in the key square, and applying the encryption rules.

Cryptography Basics: An introduction to basic cryptography concepts is beneficial. Understanding terms like encryption, decryption, keys, and ciphers lays the foundation for implementing specific algorithms like the Playfair Cipher.

History of Playfair Cipher: A brief historical overview of the Playfair Cipher is helpful. Knowing how and why the Playfair Cipher was used in the past adds context to the practical implementation.

Debugging Skills: Proficiency in debugging is essential for identifying and fixing issues in the code. Learners should be familiar with using debugging tools and techniques.

Documentation Reading and Writing: Reading and understanding documentation is a valuable skill. Learners may need to refer to language-specific documentation for functions or libraries used in their program. Additionally, documenting their own code is crucial for future reference and collaboration.

Ethical Considerations: Awareness of ethical considerations related to cryptography, such as the responsible use of encryption and privacy concerns, is important for a well-rounded understanding of the practical application.

Definition and example of Play-Fair Cipher

The Playfair Cipher is a cryptographic technique that encrypts pairs of letters (digraphs) instead of single letters as in the case of traditional ciphers. It uses a key square, which is a 5x5 matrix of letters, to perform the encryption.

Here's a step-by-step explanation of the Playfair Cipher along with an example:

Key Square Generation:

Choose a Key: Choose a keyword (e.g., "KEYWORD"). Remove any duplicate letters and combine with the remaining letters of the alphabet (excluding 'J'). In this example, let's use "KEYWORD" as the keyword.

Fill the Key Square: Fill a 5x5 matrix with the unique letters of the keyword and the remaining letters of the alphabet. Use a common rule to fill the matrix, such as left to right, top to bottom. Omit 'J' or combine 'I' and 'J' in the same cell.

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

Text Preprocessing:

Remove Spaces: Remove any spaces from the plaintext.

Example: "HELLO WORLD" becomes "HELLOWORLD".

Handle Repeated Letters: If there are repeated letters in a digraph, insert a filler letter (usually 'X') between them.

Example: "HELLOWORLD" becomes "HE LX LO WO RL D".

Encryption Logic:

Divide into Digraphs: Divide the processed text into digraphs (pairs of letters).

Example: "HE LX LO WO RL D" becomes ["HE", "LX", "LO", "WO", "RL", "D"].

Determine Positions: For each digraph, find the positions of the two letters in the key square.

Apply Encryption Rules: If the letters are in the same row, replace them with the letters to their immediate right (circular).

If the letters are in the same column, replace them with the letters immediately below (circular).

If the letters form a rectangle, replace them with the letters at the opposite corners of the rectangle.

Example: Using the key square, "HE" becomes "OL", "LX" becomes "OX", and so on.

Encrypted Text:

Combine the encrypted digraphs to form the final ciphertext.

Example: "OL OX LO RO OK TF"

So, the Playfair Cipher encryption of "HELLO WORLD" with the key "KEYWORD" is "OLOX LORO OKTF".

Procedure to be followed/Source code With Output:

Practical No-5: Write a Program to implement the Rail Fence Technique for encryption.

Objective:

The objective of using the Rail Fence Technique in a practical scenario is to encrypt or obfuscate sensitive information for secure communication.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Algorithmic Implementation
- Coding Proficiency in C/C++/Java/Python
- Cryptography Awareness
- Debugging and Troubleshooting
- User Interaction and Input Handling

Expected Course Outcomes (Cos): CO2

Practical Outcome (PRo): Write a Program to implement Rail Fence for basic encryption.

Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Critical Thinking in Security Context
- Follow ethical practices.
- Curiosity and Inquisitiveness
- Adaptability and Continuous Learning

Prerequisite Theory:

The Rail Fence Cipher is a transposition cipher that rearranges the characters of a message by writing them in a zigzag pattern across a certain number of rows. The process involves encrypting and decrypting the message. Here's a basic explanation of both encryption and decryption for the Rail Fence Cipher:

Encryption:

Choose the Number of Rails (Rows): - Decide on the number of rails (rows) to be used in the zigzag pattern. This is often referred to as the "**key**" in the context of the Rail Fence Cipher.

Write the Message in Zigzag Pattern: - Write the message in a zigzag pattern across the chosen number of rows. Start from the top-left corner and move diagonally until you reach the bottom, then reverse direction. Repeat this process until you fill the entire pattern.

Read the Encrypted Message: - Once the zigzag pattern is filled with the message, read the characters row by row from the top-left corner to the bottom-right corner. The resulting sequence is the encrypted message.

Example:

Suppose we want to encrypt the message "**INFORMATION SECURITY**" using a rail fence cipher with encryption **key 3**. Here is how we would proceed.

Plain Text: - INFORMATION SECURITY

Number of Rails/Key: - 3

i. Arrange the plain text characters in an array with 3 rows(the key determines the number of rows), forming a zig-zag pattern:

I	**N**	**F**	**O**	**R**	**M**	**A**	**T**	**I**	**O**	**N**	**S**	**E**	**C**	**U**	**R**	**I**	**T**	**Y**
I				R				I				E				I		
	N		O		M		T		O		S		C		R		T	
		F				A				N				U				Y

ii. Then concatenate the non-empty characters from the rows to obtain the **ciphertext: - IRIEINOMTOSCRTFANUY**

Decryption:

Determine the Zigzag Pattern: - The number of columns in the rail fence cipher remains equal to the length of plain-text message. And the key corresponds to the number of rails. Hence, the rail matrix can be constructed accordingly. Once we've got the matrix we can figure-out the spots where texts should be placed.

Fill in the Encrypted Message: - Fill the cipher-text row wise.

Read the Decrypted Message: -Read the characters diagonally from the top-left corner to the bottom-right corner, reversing direction when needed. The resulting sequence is the decrypted message.

Example:
Ciphertext: - IRIEINOMTOSCRTFANUY

Number of Rails/Key: - 3

*				*				*				*				*		
	*		*		*		*		*		*		*		*		*	
		*				*				*				*				*

I				R				I				E				I		
	N		O		M		T		O		S		C		R		T	
		F				A				N				U				Y

<u>Decrypted Message: -</u> INFORMATION SECURITY

It's essential to note that both the encryption and decryption processes rely on the same number of rails. Knowing the key (number of rails) is crucial for decrypting the message successfully. The Rail Fence Cipher is relatively straightforward and primarily serves educational purposes or scenarios where basic encryption is sufficient.

Procedure to be followed/Source code with Output:

Practical No- 6: Write a Program to implement the RSA algorithm for asymmetric key encryption.

Objective:

The objectives of implementing the RSA algorithm includes: Secure Data Transmission, Public Key Encryption, Digital Signatures, Key Exchange, Secure Online Transactions, Authentication and Authorization, Key Management.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Algorithmic Implementation
- Coding Proficiency in C/C++/Java/Python
- Cryptography Awareness
- Debugging and Troubleshooting
- User Interaction and Input Handling

Expected Course Outcomes (Cos): CO3

Practical Outcome (PRo): Write a Program to implement RSA algorithm for encryption.

Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Critical Thinking in Security Context
- Follow ethical practices.
- Curiosity and Inquisitiveness
- Adaptability and Continuous Learning

Prerequisite Theory:

The RSA algorithm is a widely used asymmetric cryptographic algorithm that enables secure communication and digital signatures. It was introduced by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. The security of RSA relies on the difficulty of factoring the product of two large prime numbers. Here's an overview of the key steps in the RSA algorithm:

Key Generation:

- Choose two large prime numbers, p and q.
- Compute n=pq, which is used as the modulus for both the public and private keys.
- Compute $\phi(n)=(p-1)(q-1)$, where ϕ is Euler's totient function.
- Choose a public exponent e such that $1<e<\phi(n)$ and $\gcd(e,\phi(n))=1$ (i.e., e is relatively prime to $\phi(n)$).
- Calculate the private exponent d such that $de\equiv 1 \pmod{\phi(n)}$.

The public key is (e,n), and the private key is (d,n).

Encryption:

- Represent the plaintext message as a numerical value m such that $0\leq m<n$.
- Compute the ciphertext c using the public key: $c\equiv m^e \pmod n$.
- The ciphertext c is then sent to the recipient.

Decryption:

- The recipient uses the private key to decrypt the ciphertext c.
- Compute the plaintext m using the private key: $m\equiv c^d \pmod n$.
- The numerical value m is then converted back to the original plaintext message.

The security of RSA relies on the difficulty of factoring the product of two large prime numbers (n). If an adversary can factor n, they can compute $\phi(n)$ and derive the private key (d). It's important to note that RSA is computationally expensive for large key sizes, particularly for the private key operations (decryption). Therefore, in practice, RSA is often used in combination with symmetric key algorithms, where RSA is employed for secure key exchange, and a symmetric key is used for the actual data encryption. This hybrid approach combines the strengths of both asymmetric and symmetric cryptography.

Procedure to be followed/Source code With Output:

Practical No- 7: Simulate the concept of Virtual LAN using Cisco Packet Tracer.

Objective:

Create and configure Virtual LANs (VLANs) using Cisco Packet Tracer.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6 and PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Cisco Packet Tracer tool
- Simulation of Real-world scenario
- Configuration of different devices in a simulation tool.
- Testing and Troubleshooting of Network Topology in packet tracer.

Expected Course Outcomes (Cos): CO4

Expected Affective domain Outcome (ADos)

- Critical Thinking in Security Context
- Follow ethical practices.
- Curiosity and Inquisitiveness
- Adaptability and Continuous Learning

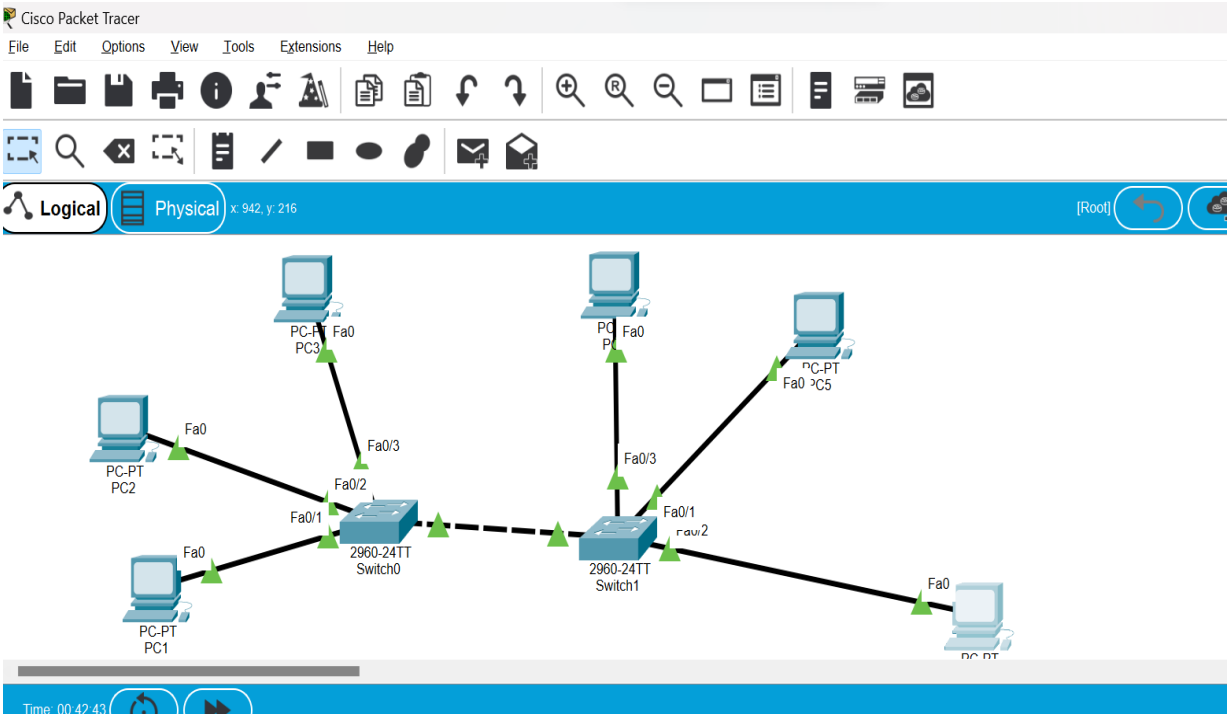
Prerequisite Theory:

VLAN is the abbreviation for Virtual LAN, i.e. Virtual Local Area Network. This is a custom network we create from one or more existing LANs. It enables a group of devices from multiple networks (both wired and wireless) to be combined into a single Logical network. The result is a VLAN that can be administered like a physical area network. The network equipment like routers or switches must support the VLAN configurations to create a VLAN.

A broadcast domain is a network segment in which if a device broadcast a packet, then all the devices in the same broadcast domain will receive it. The devices in the same broadcast domain will receive all the broadcast packets but it is limited to switches only as routers don't forward out the broadcast packet. To forward out the packets to different VLAN (from one VLAN to another) or broadcast domains, inter Vlan routing is needed. Through VLAN, different small-size sub-networks are created which are comparatively easy to handle.

Procedure to be followed/Source code:

1. **Open Cisco Packet Tracer:**
   Launch Cisco Packet Tracer and create a new blank project.
2. **Add and Connect Devices:**
   Drag and drop the devices mentioned above onto the workspace. Connect the devices using appropriate cables (e.g., Ethernet cables).



3. **Configure IP Addresses:**
   PC 1 –> 192.168.1.2 VLAN 10
   PC 2 –> 192.168.1.3 VLAN 10
   PC 3 –> 192.168.1.4 VLAN 20
   PC 4 –> 192.168.1.6 VLAN 20
   PC 5 –> 192.168.1.7 VLAN 20
   PC 6 –> 192.168.1.8 VLAN 10
4. **Create VLANs on Switches:**
   Access the CLI of each switch by clicking on it and then clicking the "CLI" tab.
   Enter the following commands to create VLANs:
   Switch0> enable
   Switch0# configure terminal
   Switch0(config)# vlan 10
   Switch0(config-vlan)# name faculty
   Switch0(config)# vlan 20
   Switch0(config-vlan)# name student
   Repeat the process on Switch1.

5. **Assign Ports to VLANs:**

Assign ports to VLANs on Switch0:

Interface fa0/1

Switch0(config)# interface range fa0/1
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 10

Interface fa0/2

Switch0(config)# interface range fa0/2
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 10

Interface fa0/3

Switch0(config)# interface range fa0/3
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 20

Assign ports to VLANs on Switch1

Interface fa0/1

Switch0(config)# interface range fa0/1
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 20

Interface fa0/2

Switch0(config)# interface range fa0/2
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 20

Interface fa0/3

Switch0(config)# interface range fa0/3
Switch0(config-if-range)# switchport mode access
Switch0(config-if-range)# switchport access vlan 10

6. **Test Connectivity:**
   - Ping between PCs in the same VLAN and between PCs in different VLANs.
   - Use the show vlan command on switches to verify VLAN configuration.
   - Use the show ip interface brief command on the router to verify sub interface configuration. Ping between devices to confirm VLAN separation.

Practical No- 8: Simulate the working of Firewall using Cisco Packet Tracer.

Objective:

Simulate the working of Firewall using Cisco Packet Tracer.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO6, PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- "Develop Applications using Java object-oriented concepts".

Expected Course Outcomes (Cos): CO4

Practical Outcome (PRo): Simulate the working of Firewall using Cisco Packet Tracer.

Expected Affective domain Outcome (ADos)

- Responsibility
- Awareness of Threats
- Follow ethical practices.

Prerequisite Theory:

A firewall is a hardware or software network security device that monitors all incoming and outgoing traffic based on a defined set of security rules, it accepts, rejects, or drops that specific traffic.

- Accept: Allow traffic.
- Reject: Block traffic but respond with “reachable error”.
- Drop: Block unanswered traffic firewall establishes a barrier between secure internal networks and untrusted external networks, such as the Internet.

Procedure to be followed:

Steps to Configure and Verify Firewall in Cisco Packet Tracer:

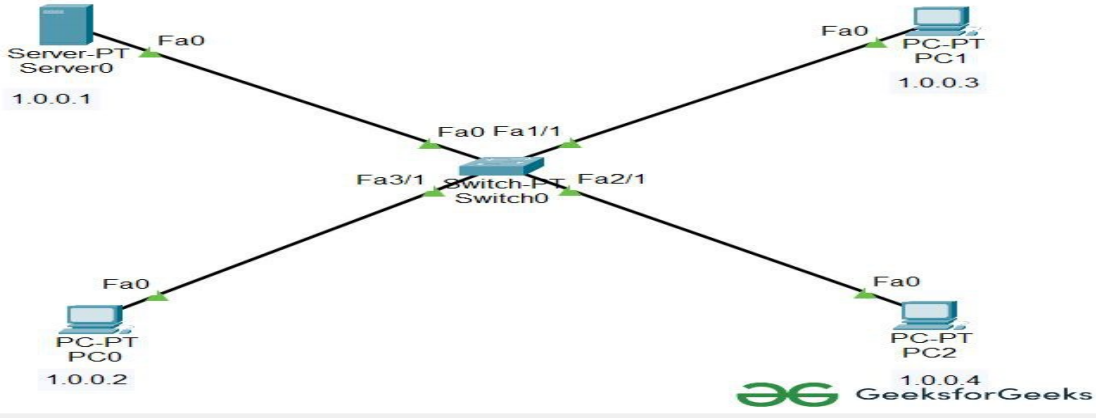
Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

Sr. No	Device	Model Name	Quantity
1	PC	PC	3
2	server	PT-server	1
3	Switch	PT-switch	1

IP Addressing Table:

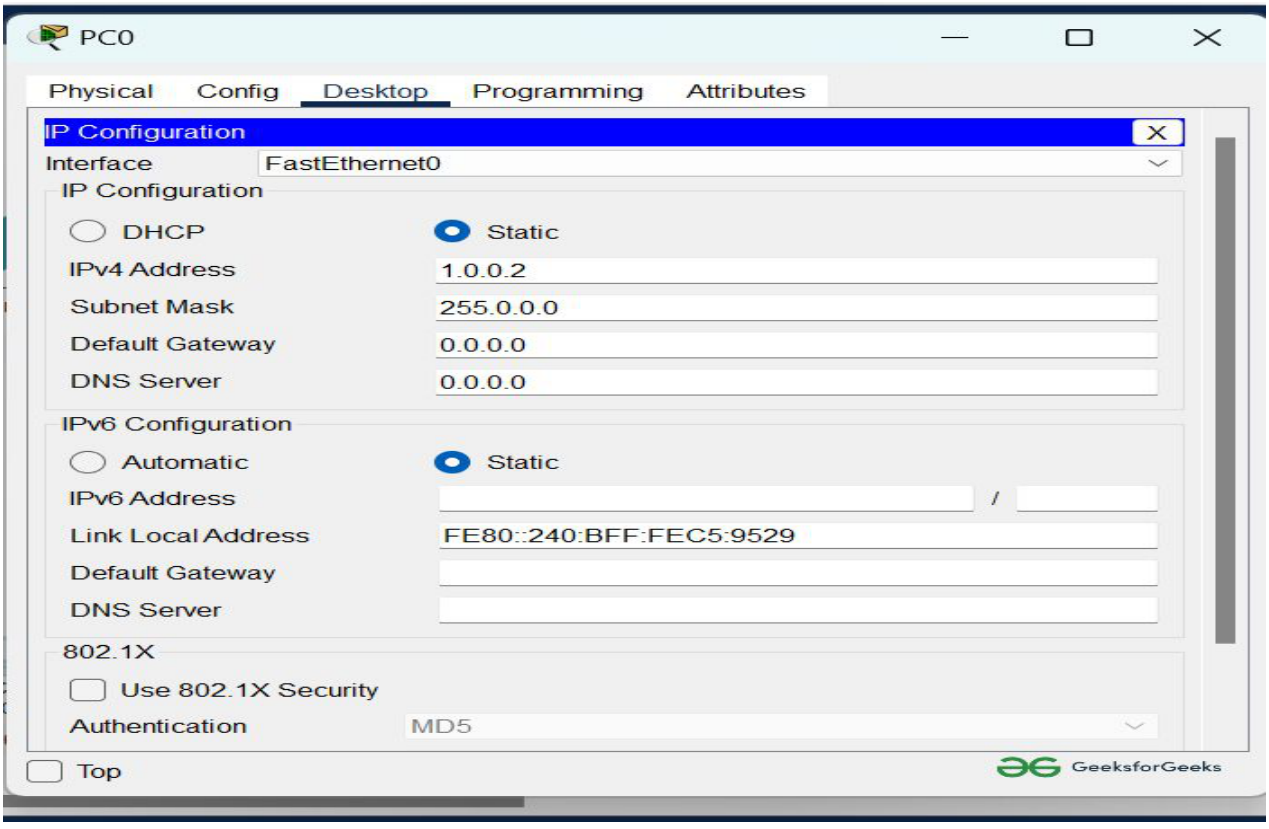
Sr. No	Device	IPv4 Address	Subnet Mask
1	Server	1.0.0.1	255.0.0.0
2	PCO	1.0.0.2	255.0.0.0
3	PC1	1.0.0.3	255.0.0.0
4	PC2	1.0.0.4	255.0.0.0

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.

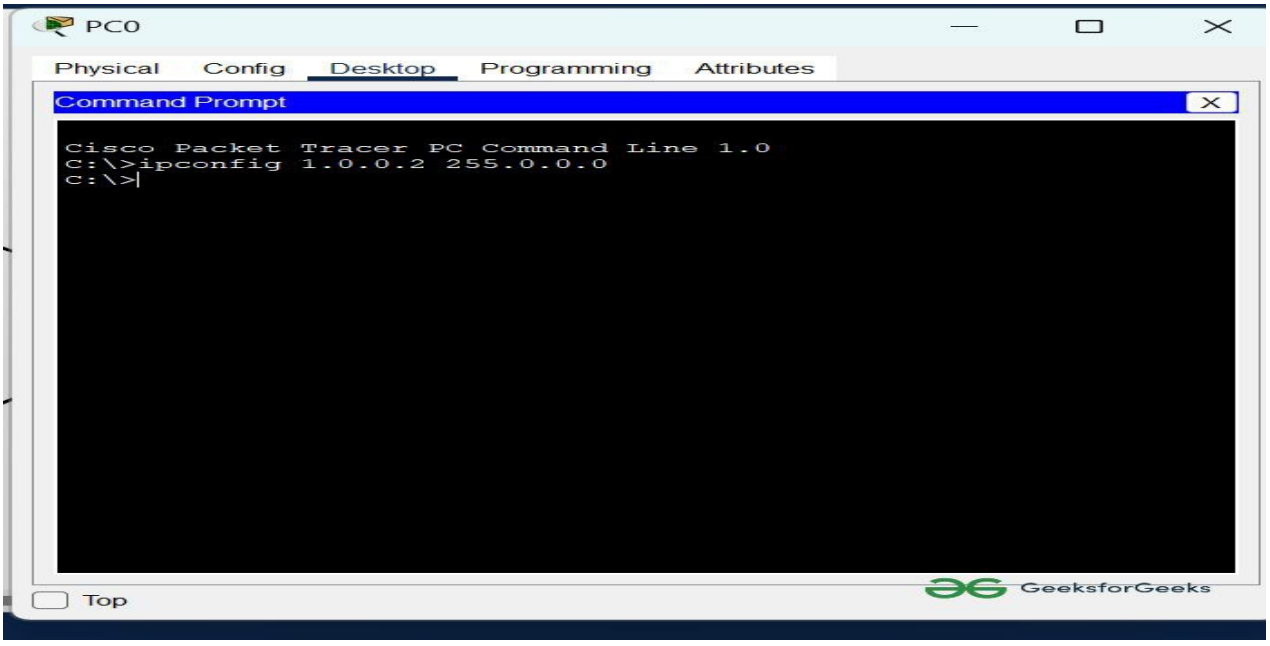


Step 2: Configure the PCs (hosts) and server with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.
- Repeat the same procedure with the server



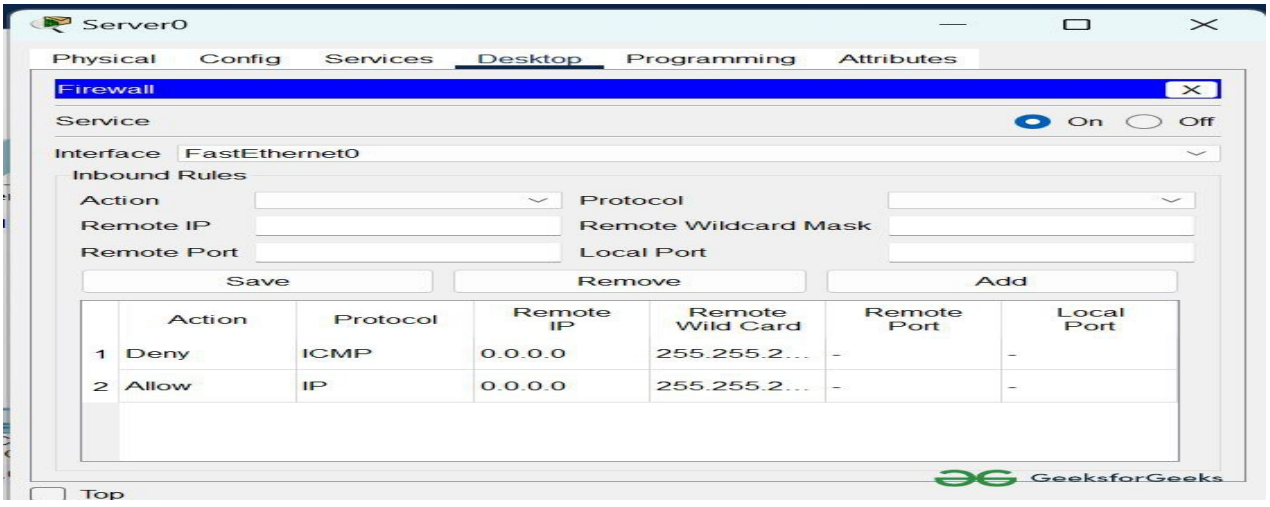
- Assigning an IP address using the ipconfig command, or we can also assign an IP address with the help of a command.
- Go to the command terminal of the PC.
- Then, type iPConfig <IPv4 address><subnet mask><default gateway>(if needed) Example: ipconfig 1.0.0.2 255.0.0.0



- Repeat the same procedure with other PCs to configure them thoroughly.

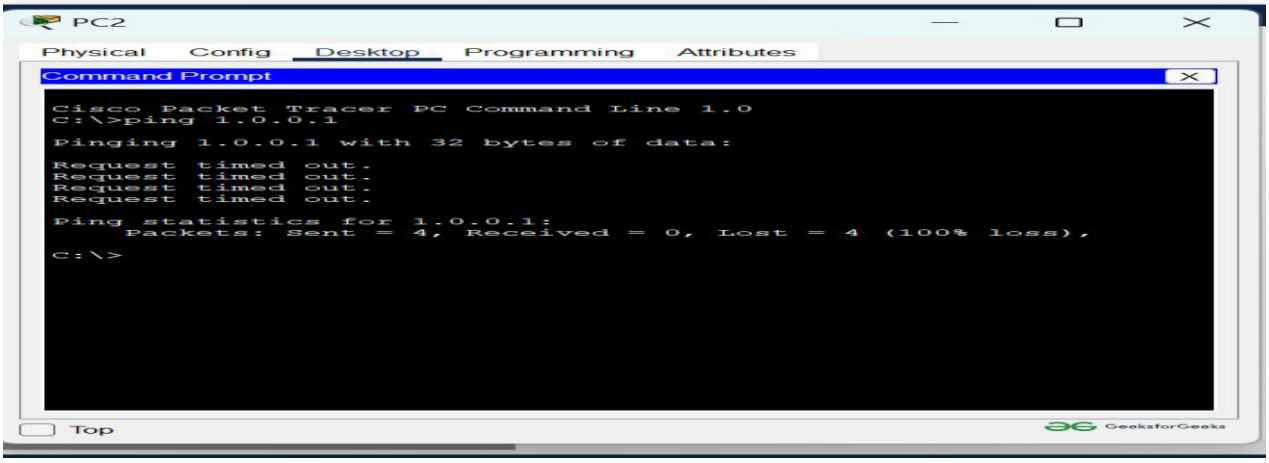
Step 3: Configuring the firewall in a server and blocking packets and allowing web browser.

- Click on server0 then go to the desktop.
- Then click on firewall IPv4.
- Turn on the services.
- First, Deny the ICMP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
- Then, allow the IP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
- And add them.



Step 4: Verifying the network by pinging the IP address of any PC.

- We will use the ping command to do so.
- First, click on PC2 then Go to the command prompt.
- Then type ping <IP address of targeted node>.
- We will ping the IP address of the server0.
- As we can see in the below image we are getting no replies which means the packets are blocked.



Check the web browser by entering the IP address in the URL.

- Click on PC2 and go to the desktop then web browser.

Practical No- 9: Study cyber security fundamentals, including common threats and mitigation strategies.

Objective:

Study cyber security fundamentals, including common threats and mitigation strategies.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO5, PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Conducting penetration tests, identifying vulnerabilities, and exploiting them.
- Exploiting vulnerabilities to gain unauthorized access.
- Configuring firewalls and intrusion detection/prevention systems.
- Analysing network traffic for signs of malicious activity.
- Perform system administration tasks on a Linux-based environment

Expected Course Outcomes (Cos): CO5

Practical Outcome (PRo)

- Identify vulnerabilities in systems and networks
- Test wireless networks
- Mitigate strategies

Expected Affective domain Outcome (ADos)

Workforce capable of preventing and mitigating cyber-attacks.
Follow ethical practices.

Prerequisite Theory:

Cybersecurity protects computer systems, networks, and data from unauthorised access, damage, or theft in the digital world. It involves measures and technologies designed to safeguard information and prevent malicious activities.

Cybersecurity aims to ensure data and systems' confidentiality, integrity, and availability, keeping them safe from unauthorized access, manipulation, or disruption. Techniques like encryption, firewalls, antivirus software, and user authentication are used to establish barriers and secure digital assets from potential risks.

Here are some common types of threats:

Malware:

(1) **Viruses:** Programs that attach themselves to legitimate programs and replicate when those programs are executed.

(2) **Worms:** Self-replicating programs that spread across networks without needing a host file.

(3) **Trojans:** Malicious programs disguised as legitimate software.

Phishing:

Attempts to trick individuals into providing sensitive information, such as passwords or credit card numbers, by posing as a trustworthy entity.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks:

Overwhelming a system, network, or website with traffic to make it unavailable or slow down its performance.

Cyber Security Fundamentals:

Confidentiality, Integrity, and Availability (CIA Triad):

- Confidentiality: Ensuring that information is only accessible to authorized individuals.
- Integrity: Maintaining the accuracy and trustworthiness of data.
- Availability: Ensuring that systems and data are accessible when needed.

Authentication and Authorization:

- Authentication: Verifying the identity of users or systems.
- Authorization: Granting appropriate access privileges to authenticated users.

Encryption:

- Using algorithms to convert data into a secure format that can only be accessed with the right decryption key.

Firewalls:

- Implementing firewalls to monitor and control incoming and outgoing network traffic based on predetermined security rules.

Intrusion Detection and Prevention Systems (IDPS):

- Monitoring and analysing network or system events to detect and respond to security incidents.

Common Threats:

Malware:

- Software designed to harm or exploit systems, including viruses, worms, and ransomware.

Phishing:

- Deceptive attempts to obtain sensitive information by pretending to be a trustworthy entity.

Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks:

- Overloading a system or network to disrupt its normal functioning.

Social Engineering:

- Manipulating individuals to disclose confidential information or perform actions against their best interests.

Insider Threats:

- Security risks that originate from within an organization, such as employees with malicious intent or unintentional mistakes.

Mitigation Strategies:

Regular Software Updates:

- Keeping operating systems, applications, and security software up-to-date to patch known vulnerabilities.

Employee Training:

- Educating users about cybersecurity best practices, including recognizing phishing attempts and avoiding suspicious links.

Access Control:

- Implementing least privilege principles to restrict access to the minimum necessary for users to perform their tasks.

Network Segmentation:

- Dividing networks into segments to limit the spread of attacks and contain potential breaches.

Backup and Recovery:

- Regularly backing up critical data and having a robust recovery plan to minimize the impact of ransomware or data loss.

Incident Response Plan:

- Developing and regularly testing a plan to respond to and recover from cybersecurity incidents.

Security Audits and Monitoring:

- Conducting regular security audits and monitoring network and system activities for signs of unusual behaviour.

Endpoint Security:

- Securing individual devices (endpoints) through antivirus software, firewalls, and other protective measures.

Write answers for following questions:

1. List Cyber Attacks and Explain any Four.
2. List Vulnerabilities in Cyber Security.
3. Define Hackers, Explain Its Types.
4. Define Hacking and List Methods of Hacking.
5. List Attack Vectors and Explain any Four.

Practical No- 10: Study of Kali Linux Operating System for cybersecurity.

Objective:

Study of Kali Linux Operating System for cybersecurity.

Expected Program Outcomes (POs): PO1, PO2, PO3, PO4, PO5, PO7

Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Run kali Linux command line tools
- Identify and assess vulnerabilities in systems and networks.
- Secure wireless networks

Expected Course Outcomes (Cos): CO5

Practical Outcome (PRo)

- Working of kali Linux

Expected Affective domain Outcome (ADos)

Ethical awareness
Problem solving attitude
Risk awareness

Prerequisite Theory:

Kali Linux is a popular operating system designed for penetration testing, ethical hacking, and cybersecurity tasks. It comes pre-installed with a variety of tools that are useful for security professionals, researchers, and enthusiasts. Here is a guide on how to study Kali Linux for cybersecurity:

1. Understand the Basics of Linux:
    - Familiarize yourself with basic Linux commands and the file system structure.
    - Learn about package management using tools like apt or dpkg.
2. Install and Set Up Kali Linux:
    - Download the latest version of Kali Linux from the official website.
    - Install it on a virtual machine or as a dual boot with your existing operating system.
3. Explore Kali Tools:
    - Familiarize yourself with the various tools available in Kali Linux. These include tools for information gathering, vulnerability analysis, wireless attacks, exploitation, post-exploitation, and more.
    - Tools such as nmap, Wireshark, Metasploit, Burp Suite, and Aircrack-ng are commonly used.

4. Learn Networking Concepts:
    - Understand networking fundamentals, including TCP/IP, subnetting, and protocols.
    - Learn how to use tools like netstat, tcpdump, and Wireshark for network analysis.
5. Study Web Application Security:
    - Learn about common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
    - Use tools like OWASP Zap and Burp Suite for web application testing.
6. Practice Ethical Hacking:
    - Set up a lab environment for practicing ethical hacking. This can include intentionally vulnerable machines or platforms like Hack The Box or TryHackMe.
    - Practice different types of attacks, such as password cracking, privilege escalation, and social engineering.
7. Understand Exploitation Techniques:
    - Study common exploitation techniques and methodologies.
    - Learn how to use tools like Metasploit for automated exploitation.
8. Study Forensics and Incident Response:
    - Explore tools and techniques for digital forensics and incident response.
    - Understand how to analyze logs, conduct memory forensics, and investigate security incidents.

Kali Linux Operating System Commands

1. System Information:
    - uname -a: Display system information.
    - lsb_release -a: Display distribution-specific information.
2. Package Management:
    - apt update: Update package lists.
    - apt upgrade: Upgrade installed packages.
    - apt install [package]: Install a package.
    - apt remove [package]: Remove a package.
    - apt search [keyword]: Search for a package.
3. File and Directory Commands:
    - ls: List files and directories.
    - cd [directory]: Change directory.
    - pwd: Print current working directory.
    - cp [source] [destination]: Copy files or directories.
    - mv [source] [destination]: Move or rename files or directories.
    - rm [file]: Remove (delete) a file.
    - mkdir [directory]: Create a new directory.

4. Text Manipulation:
    - cat [file]: Display the contents of a file.
    - nano [file] or vim [file]: Open a text editor.
    - grep [pattern] [file]: Search for a pattern in a file.
    - echo [text] > [file]: Redirect text to a file.
    - head [file] and tail [file]: Display the first/last lines of a file.
5. User and Permission Management:
    - whoami: Display the current username.
    - sudo [command]: Execute a command with superuser privileges.
    - useradd [username]: Add a new user.
    - passwd [username]: Set or change a user's password.
    - chown [user:group] [file]: Change ownership of a file.
    - chmod [permissions] [file]: Change permissions of a file.
6. Networking Commands:
    - ifconfig or ip addr: Display network interfaces and their configurations.
    - ping [hostname or IP]: Test network connectivity.
    - traceroute [hostname or IP]: Display the route packets take to a network host.
    - netstat -tulpn: Display listening ports.
    - iwconfig: Display wireless network information.
7. Security and Penetration Testing Tools:
    - Many security tools come pre-installed in Kali. Examples include:
    - nmap: Network scanning tool.
    - msfconsole: Metasploit Framework console.
    - aircrack-ng: Wireless security tool.
    - john: Password cracker.
    - hydra: Password brute-force tool.
8. System Services:
    - systemctl start/stop/restart [service]: Manage system services.
    - service [service] start/stop/restart: Legacy service management.
9. System Updates:
    - apt update: Update package lists.
    - apt upgrade: Upgrade installed packages.
10. System Shutdown/Reboot:
    - shutdown now: Shutdown the system immediately.
    - reboot: Reboot the system.

Write answers for following questions

1. Why is Kali Linux used for cyber security?
2. List and explain different tools available in the Kali Linux.