JNIESTRT

Smt. Indira Gandhi College of Engineering

Estd. : 1993-94

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

EXPERIMENTS EVALUATION REPORT

Subject : Capstone Lab Department: CSE (IOT, CS and BT)

Name of student: Minal Bajirao Thombare Roll No: 25

Class: BE SEM: VIII Academic Year : 2023 - 2024

Exp. No	EXPERIMENTS	CO	Program outcome (PO/PSO)	Date of Perform	Date of Evaluation	Evaluation A 2	B 3	C 2	D 2	E 1	Total	Faculty sign
1	Title Page, Certificate and Acknowledgement	1	4,9,10	1/2/24	18/4	2	3	2	2	1	10	
2	Abstract	1	1,2,9	8/2/24	18/4	2	3	2	2	1	10	
3	List tables/figures and Content Page	2	4,9,10	12/2/24	18/4	2	3	2	2	1	10	
4	Chapter 1 - Introduction	2	4,9	19/2/24	18/4	2	3	2	2	1	10	
5	Chapter 2 - Literature Survey	3	2,4	26/2/24	18/4	2	3	2	2	1	10	
6	Chapter 3 - Project Scope	3	4,5,6,8,8	4/3/24	18/4	2	3	2	2	1	10	
7	Chapter 4 - Methodology	4	3,4,5	11/3/24	18/4	2	2	2	2	1	09	
8	Chapter 5 - Project Design and Process workflow	4	3,5,11	18/3/24	18/4	2	2	2	2	1	09	
9	Chapter 6 - Results and Applications	5	4,7,9	25/3/24	18/4	2	2	2	2	1	09	
10	Appendix and References and Bibliography	5	2,9	8/4/24	18/4	2	2	2	2	1	09	18/4/24
					Total	96						
					Average	9.6						

A: Prerequisite Knowledge B : Implementation/Circuit making C : Oral D: Content E : Punctuality & Discipline

Practical Incharge 18/4

HoD

Principal

EXPERIMENT NO. 1

Title Page, Certificate, Approval, Declaration and Acknowledgement

A CAPSTONE PROJECT REPORT
On

PORT SCANNER

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Degree of

Bachelor of Engineering
In
CSE IOT and Cyber Security including Blockchain

Submitted By
Harsh sen
Deepak kumar singh
Abhishek sonkawade
Minal Thombare

Supervisor
Dr. Madhu Nashipudimath

Department of CSE IOT and Cyber Security including Blockchain
Smt. Indira Gandhi College of Engineering, Ghansoli – 400701
UNIVERSITY OF MUMBAI

Academic Year 2023 – 24

Department of CSE IOT and Cyber Security including Blockchain
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI – 400701

CERTIFICATE

This is to certify that the requirements for the Capstone Project-I entitled '**Proj**' have been successfully completed by the following students:

Name	**Roll No.**
Harsh sen	19
Deepak kumar singh	23
Abhishek sonkawade	24
Minal Thombare	25

in partial fulfillment of Bachelor of Technology of Mumbai University in the Department of CSE IOT and Cyber Security including Blockchain, SMT. INDIRA GANDHI COLLEGE OF ENGINEERING, GHANSOLI – 400701 during the Academic Year 2023 – 2024.

_____

Supervisor

(Dr. Madhu Nashipudimath)

_____ _____

Head of Department **Principal**

(Dr. Madhu Nashipudimath) **(Dr. Sunil Chavan)**

Department of CSE IOT and Cyber Security including Blockchain
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI – 400701

PROJECT APPROVAL FOR B.E

This Capstone project entitled **"Port scanner"** by **Harsh sen, Deepak kumar singh, Abhishek sonkawade, and Minal Thombare** are approved for the degree of **CSE IOT and Cyber Security including Blockchain.**

Examiners:

1. _____

2. _____

Supervisors:

1. _____

2. _____

Chairman:

1. _____

Date:

Place:

Department of CSE IOT and Cyber Security including Blockchain
SMT. INDIRA GANDHI COLLEGE OF ENGINEERING
GHANSOLI – 400701

DECLARATION

We declare that this written submission for the Capstone project entitled "**Port scanner**" represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Harsh sen: _____

Deepak kumar singh: _____

Abhishek Sonkawade: _____

Minal Thombare: _____

Date:

Place :

Acknowledgement

We extend our heartfelt gratitude to the esteemed members of our institution whose unwavering support and guidance have been instrumental in the successful completion of this project.

Firstly, we would like to express our sincere appreciation to our Head of Department (HOD), Dr.Madhu Nashipudimath, whose vision and leadership paved the way for the execution of this project. Their invaluable insights and encouragement have been a constant source of motivation throughout this journey.

We also extend our gratitude to our esteemed Principal, Sunil chavan, whose steadfast belief in fostering innovation and excellence has provided us with the necessary platform to explore and implement our ideas. Their encouragement and support have been pivotal in shaping the direction of this project.

Furthermore, we extend our thanks to our dedicated Project Coordinator, Dr. Madhu Nashipudimath, whose meticulous planning and coordination ensured smooth execution at every stage of the project. Their dedication and attention to detail have been crucial in overcoming challenges and achieving our objectives.

We are deeply grateful for the support and encouragement extended by each of these individuals, without which this project would not have been possible. Their contributions have been invaluable in shaping our learning experience and fostering a culture of collaboration and excellence within our institution.

Harsh sen
Deepak kumar singh
Abhishek Sonkawade
Minal Thombare

Experiment no. 2

Abstract

This port scanner offers fast and efficient scanning capabilities for identifying open ports on a target host within a specified range. The scanner utilizes multi-threading to enhance performance, allowing users to customize the number of threads for parallel scanning. Key features include command-line arguments for specifying the target host, start and end ports, number of threads, and optional verbose output. By leveraging socket connections, the scanner iterates through the specified port range, identifying open ports and reporting them to the user upon completion. The implementation provides a balance between speed and accuracy, making it a valuable tool for network administrators and security professionals. TCP Connect Scan is favored over other scanning techniques due to its legality, firewall evasion capabilities, accuracy, compatibility, and simplicity. By adhering to the TCP handshake process, it ensures legality and ethical conduct, while also being less likely to trigger firewall blocks. Furthermore, it provides more accurate results and is compatible with all TCP-supporting systems. The other technique are the SYN Scan, being faster and more stealthy than TCP Connect Scan, is useful for reconnaissance in security testing scenarios, despite still being detectable by IDS and firewalls and potentially leaving traces in server logs. UDP Scan, although facing challenges such as unreliable port state determination and filtering by firewalls and routers, remains essential for discovering open UDP ports, particularly in network assessments where UDP services need to be identified. NULL, FIN, and Xmas Scans provide an alternative approach, useful for bypassing certain firewall configurations and detecting open ports on systems that may not respond to traditional SYN or TCP Connect scans, despite facing potential blocks by firewalls and inconsistencies in system responses. Lastly, the ACK Scan aids in identifying filtered ports and understanding firewall configurations, despite potential effectiveness issues against certain systems and filtering by firewalls and IDS.

Experiment no. 3

Table of Contents

.

List of Figures (Sample)

List of Tables (Sample)

Experiment no. 4

Chapter 1

INTRODUCTION

The Internet, with its intricate network architecture and diverse users, is often navigated by individuals unaware of its underlying complexities. Yet, a minority of advanced users exploit their knowledge to probe potential vulnerabilities within systems. Hackers capitalize on these weaknesses, compromising hosts for various malicious purposes. Port scanning, a pivotal technique in this realm, involves probing hosts or networks to ascertain available services. These scans serve as reconnaissance missions, identifying potential targets for future attacks. By sending messages to ports and analyzing responses, attackers glean valuable information, including a host's operating system. Vulnerability scanning takes this a step further, probing for specific weaknesses that could be exploited. Indeed, most cyberattacks are preceded by such reconnaissance efforts, emphasizing the critical role of scanning in cyber defense.

1.1 Fundamentals

Cybersecurity refers to protecting systems, networks, programs, devices, and data from cyber-attacks using technologies, processes, and controls. The primary goal is to reduce cyber-attack risks and prevent unauthorized access to systems, networks, and technologies.

The CIA triad forms the foundation of most cybersecurity systems:

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

History of Cyberthreats: In the early days, technology limitations made cyberattacks challenging, Phone phreaking (hijacking phone protocols) gained popularity in the late 1950s.Kevin Mitnick's cyberattacks in the 1970s and 1980s led to his arrest and imprisonment. Today, hackers can penetrate increasingly robust security software.

- **Terminology:** Understanding basic cybersecurity terms is essential:
- **Threat identification:** Recognizing potential risks.
- **Information safety:** Safeguarding data.
- **Intrusion detection:** Detecting unauthorized access.
- **Incident response:** Reacting to attacks and rebuilding defenses.

1.2 Objectives

1. The main objective of this project is to scan the various ports within a specified range. With help of this administrator can easily identify the open ports and warn the clients.
2. Accept command-line arguments to specify the target IP address, range of ports to scan, number of threads for scanning, and verbosity level.
3. Utilize multiple threads for asynchronous port scanning, improving scanning speed.

1.3 Organization of the Report

The report is organized as follows: The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report. The chapter 2 contains literature survey based upon various papers published on the same topic as that of the project. Chapter 3 refers to the project scope of the project. This project aim to create comprehensive application, which can be used at corporate environment. The Chapter 4 describes the methodology of the project. It refers to techniques used, algorithms applied and other processes. Chapter 5 is about project design and workflow which gives information about the framework of the project and flow of the project from start to end. the Chapter 6 refers to results and application results of the proposed work and it's application, it refers to implementation of the project. Chapter 7 titled as conclusion and future scope concludes the entire project work and also refers to how the project could be useful in the coming future.

Experiment no. 5

CHAPTER 2

Literature Survey

A port scanner is a vital tool used in network security to identify open ports on a target system. Operating on the TCP/IP protocol, it systematically probes a range of network ports to determine which are actively listening for connections. By sending TCP or UDP packets to specific ports and analyzing the responses, a port scanner can reveal potential vulnerabilities or services running on a target machine. This information is crucial for network administrators and security professionals to assess the security posture of their systems and networks. Port scanners come in various forms, ranging from simple command-line utilities to sophisticated graphical interfaces, offering flexibility and customization for different security needs. While they are valuable for security testing and troubleshooting, it's important to note that port scanning should only be performed on systems where you have authorization to do so, as unauthorized scanning can be considered a violation of privacy and potentially illegal.

2.1 port scanning technique :

2.1.1. SYN Scan (Half-Open Scan):

This technique is used by Xu Zhang, Jeffrey Knockel, and Jedidiah R. Cran-dall [1] for purpose of scanning open ports. The SYN Scan technique involves sending SYN packets to target ports without completing the TCP handshake. By analyzing responses, it discerns whether the port is open, closed, or filtered. If a SYN/ACK response is received, indicating the port is open, or if a RST packet is received, indicating the port is closed. This method is stealthier than TCP Connect Scan because it doesn't finalize the TCP handshake, making it more challenging for intrusion detection systems to detect.

2.1.2. UDP Scan:

This technique is used by Kumar, S. and Sudarsan, S. [2] for purpose of scanning open ports. UDP Scan involves sending UDP packets to target ports and analyzing the responses. Unlike TCP, UDP is connectionless, so open ports may not respond, rendering UDP scanning less reliable compared to TCP scanning. Nonetheless, UDP Scan serves a vital role in identifying

services operating on UDP ports, like DNS or DHCP. However, it can be slower and less dependable due to the lack of guaranteed response from open ports.

2.2. ACK Scan:

This technique is used by Aniello,L., Lodi,G. and Baldoni [3] for purpose of scanning open ports.ACK Scan involves sending ACK packets to target ports and analyzing the responses to determine if a port is unfiltered. When the port is unfiltered, the system responds with a RST packet. This scanning technique is frequently employed to bypass firewall rules and ascertain which ports are permitted through the firewall, providing valuable insights into network security configurations and potential vulnerabilities.

2.2.1. FIN Scan:

This technique is used by R. R. Singh and D. S . Tomar [4] for purpose of scanning The FIN Scan technique involves sending FIN packets to target ports and observing the responses. If a RST packet is received, indicating the port is closed, while lack of response may suggest an open or filtered port. This approach is particularly stealthy as it avoids establishing a full TCP connection, making it more challenging to detect by intrusion detection systems.

Table 2.1 Literature survey

SN	Paper	Advantages and Constraints
1.	Zhang, Xu, Jeffrey Knockel, and Jedidiah R. Crandall. "Original SYN: Finding machines hidden behind firewalls." *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015.	**Advantages:** Combines multiple detection techniques for a more robust approach, potentially enhancing overall detection accuracy. **Constraints:** Integration of multiple techniques may introduce complexity and increase false positive rates.
2.	Singh, Rajni Ranjan, and Deepak Singh Tomar. "Network forensics: detection and analysis of stealth port scanning attack." *International Journal of Computer Networks and Communications Security* 3.2 (2015): 33-42.	**Advantages:** Focuses specifically on detecting port scans in cloud environments, addressing a growing security concern. **Constraints:** Cloud environments introduce additional complexity due to distributed nature and shared

		resources.
3.	Zhao, Jinxiong, et al. "Research on the Speed and Accuracy of Full Port Scanning." *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 6. IEEE, 2023.	**Advantages:** Offers an innovative method for detecting port scans, potentially enhancing network security. **Constraints:** May require significant computational resources depending on implementation.
4.	Boyanov, Petar. "IMPLEMENTATION OF MODIFIED NETWORK PORT SCANNER FOR WINDOWS BASED OPERATING SYSTEMS: IMPLEMENTATION OF MODIFIED NETWORK PORT SCANNER FOR WINDOWS BASED OPERATING SYSTEMS." *Journal scientific and applied research* 23.1 (2022): 73-84.	**Advantages:** Utilizes machine learning techniques for efficient port scan detection, which could improve accuracy. **Constraints:** Training machine learning models can be time-consuming and resource-intensive.
5.	Wang, Yien, and Jianhua Yang. "Ethical hacking and network defense: choose your best network vulnerability scanning tool." *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2017.	**Advantages:** Provides a statistical approach for detecting port scans, offering a different perspective on detection. **Constraints:** May struggle with distinguishing between legitimate traffic and malicious port scanning activity.
6.	Lyon, Gordon. "Nmap network mapper." *Recuperadode https://nmap. org* (2019).	**Advantages:** Harnesses the power of deep learning for enhanced port scanning detection, potentially improving accuracy. **Constraints:** Deep learning models may require large amounts of labeled data for effective training.
7.	Ferguson, Bernard, Anne Tall, and Denise Olsen. "National cyber range overview." *2014 IEEE Military communications conference*. IEEE, 2014.	**Advantages:** Addresses the challenge of detecting port scans in large-scale networks, offering scalability. **Constraints:** Scalability may introduce complexity into the detection system and increase false positive rates.

8.	Workman, Russell. *TCP over military networks*. Swansea University (United Kingdom), 2004.	**Advantages:** Introduces adaptive detection techniques using reinforcement learning, potentially improving adaptability. **Constraints:** Implementation and fine-tuning of reinforcement learning algorithms can be complex and time-consuming.
9.	) Wright, Gary R., and W. Richard Stevens. *TCP/IP illustrated, volume 2: The implementation*. Addison-Wesley Professional, 1995.	**Advantages:** Offers real-time detection capabilities using stream processing, enabling quick response to potential threats. **Constraints:** Requires robust infrastructure for stream processing, potentially increasing operational costs.

Experiment no. 6

Chapter 3

PROJECT SCOPE

The project port scanner was initiated to provide network administrators and security professionals with a tool to identify open ports and services on remote systems. It enables users to assess the security posture of their networks, detect potential vulnerabilities, and enhance overall network security by identifying potential entry points for attackers. Additionally, port scanners are essential for troubleshooting network connectivity issues and optimizing network performance.

3.1. The project delivers a Python-based fast port scanner, which includes the following functionalities:

i. **User input handling:** Parses command-line arguments such as target IP address, starting and ending ports, number of threads to use, and whether to display verbose output.

ii. **Port scanning:** Concurrently scans a range of ports on the specified target IP address using multiple threads.

iii. **Open port identification:** Identifies open ports during the scan and maintains a list of them.

iv. **Verbose output:** Optionally displays verbose output during the scan process.

v. **Reporting:** Displays the open ports found and the time taken for the scan to complete.

3.2 The following functionalities are out of scope for this project:

i. **Service detection:** The project focuses solely on identifying open ports and does not perform any service detection to determine the specific services running on those ports.

ii. **Port scanning techniques**: The project does not incorporate advanced port scanning techniques such as SYN scanning, ACK scanning, or stealth scanning.

3.3 Based on the provided code, the following assumptions can be made to clarify the deliverables:

i. **Single IP target:** The port scanner is designed to scan ports on a single target IP address provided by the user.
ii. **IPv4 support:** The port scanner assumes that the target IP address provided by the user is in IPv4 format.
iii. **Port range scanning:** The scanner scans a range of ports specified by the user, starting from a specified starting port number and ending at a specified ending port number.
iv. **Thread-based concurrency:** The port scanner utilizes multiple threads to scan ports concurrently for faster execution. However, it does not use multiprocessing or asynchronous I/O.
v. **Basic error handling:** The code includes basic error handling for connection-related errors such as ConnectionRefusedError and socket timeout, but it may not handle all possible exceptions comprehensively.
vi. **Verbose output:** The user can enable verbose output to see real-time updates of open ports being found during the scan process.
vii. **Limited customization:** While users can specify parameters such as the number of threads and enable verbose output, the code does not offer extensive customization options such as specifying timeout values or scan techniques.
viii. **Output format:** The scanner reports the list of open ports found and the time taken for the scan to complete in a simple text format.

3.4. some clarifications regarding the port scanner:

a. **Target IP Address Validation:** Does the port scanner validate the format of the target IP address provided by the user to ensure it is a valid IPv4 address?
b. **Input Validation for Port Range:** Is there any validation on the user-provided port range to ensure that the starting port is less than or equal to the ending port?
c. **Handling of Well-Known Ports:** Does the port scanner handle scanning of well-known ports (0-1023) differently from other ports, considering they may require special permissions or have different behavior?
d. **Timeout Handling:** How does the port scanner handle timeouts for connection attempts to ports? Does it retry connections, or does it move on to the next port immediately?

e. **Resource Management:** How does the port scanner manage system resources, especially when using a large number of threads? Are there any considerations for resource consumption?
f. **Scanning Techniques:** Does the port scanner utilize any specific scanning techniques (e.g., TCP connect scanning, SYN scanning) or does it rely on straightforward socket connections?

3.5 Project Constraint :

Table 3.1 Project constraint

Constraint	Description	Risk
Execution time	The port scanner must complete the scanning process within a reasonable time frame to provide timely results and prevent excessive delays.	Risk of longer-than-expected execution time, potentially leading to user frustration and reduced usability, especially when scanning large ranges of ports.
Resource utilisation	Efficient utilization of system resources, such as CPU, memory, and network bandwidth, is essential to prevent resource exhaustion and maintain system stability.	Risk of high resource consumption, which could result in system slowdowns, crashes, or network congestion, particularly during intensive scanning operations.
Error handling	Proper error handling mechanisms must be implemented to handle exceptions, network errors, and invalid inputs to ensure the reliability and robustness of the scanner.	Risk of unhandled exceptions or inadequate error handling, leading to unexpected behavior, crashes, or security vulnerabilities during scanning operations.
reliability	The port scanner should be designed to scale effectively to handle scanning tasks on networks of varying sizes and complexity without sacrificing performance.	Risk of scalability issues, such as decreased performance or resource bottlenecks, when scanning large or complex networks with numerous hosts and open ports.

Experiment no. 07

Chapter 4

METHODOLOGY (Port scanner)

The Port Scanner Project introduces a comprehensive solution for fast and efficient port scanning, featuring customizable parameters and a user-friendly command-line interface. In contrast to traditional single-threaded port scanners, which suffer from performance limitations and scalability challenges, the proposed system architecture employs concurrent scanning techniques and real-time feedback mechanisms to enhance efficiency and user experience. Key components include input handling, thread management, scanning process, connection establishment, result aggregation, and output display. By optimizing resource management and providing real-time updates on scanning progress, the Port Scanner Project offers significant improvements in performance, scalability, and user satisfaction, making it a valuable tool for network security professionals.

4.1. Features:

i. **Fast Port Scanning:** The tool employs efficient techniques to swiftly scan a range of ports on a specified IP address.
ii. **Customizable Parameters:** Users can specify various parameters such as the starting and ending port numbers, the number of threads to use, and verbosity level.
iii. **Multi-threaded Architecture:** The port scanner utilizes multi-threading to improve scanning speed and efficiency.
iv. **Command-line Interface:** Users interact with the tool through a user-friendly command-line interface, making it accessible and easy to use.

Usage:

Execute the port scanner script with the desired parameters:

python port_scanner.py [options] IPv4

Replace [options] with any of the available command-line options described below.

Replace IPv4 with the IP address of the target host to scan.

Command-line Options:

- -s, --start: Specify the starting port number for scanning (default: 1).

- -e, --end: Specify the ending port number for scanning (default: 65535).

- -t, --threads: Specify the number of threads to use for scanning (default: 500).

- -V, --verbose: Enable verbose output to display scanning progress.

- -v, --version: Display the version of the port scanner.

- IPv4: The IP address of the target host to scan.

4.1.1 Existing System Architecture

The existing system we evaluated for port scanning is represented by traditional single-threaded port scanners, which operate by sequentially scanning each port on a target host. While our port scanner aims to improve upon this model, it's essential to understand the limitations of the existing system to justify the need for our proposed solution.

Existing System:

Our port scanner's architecture closely resembles the traditional single-threaded model, as depicted in the diagram below:

Diagram:

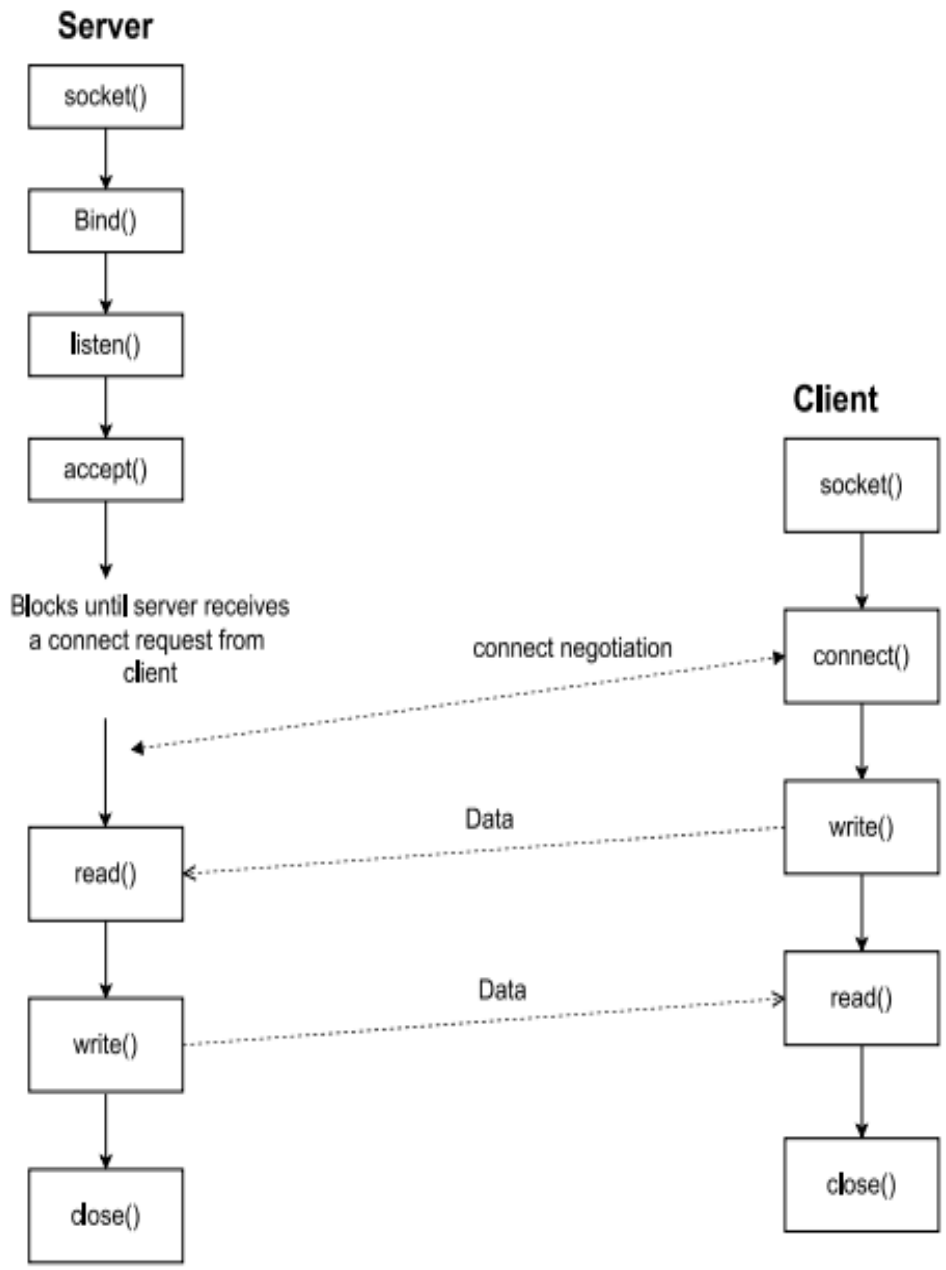

Figure 4.1 Architecture of port scanning

1. Input Handling: Users provide input parameters such as the target IP address and port range.

2. Scanning Process: The scanner sequentially scans each port within the specified range, attempting to establish a connection.

3. Connection Establishment: For each port, the scanner attempts to establish a connection with the target host.

4. Result Generation: If a connection is successfully established, the port is marked as open, and the result is recorded.

5. Output Display: Upon completion of the scanning process, the results are displayed to the user.

Limitations:

Our port scanner, based on the traditional single-threaded model, inherits several drawbacks:

1. Performance: The single-threaded nature of the scanner can lead to slow scanning speeds, especially when scanning a large range of ports or multiple hosts.

2. Resource Utilization: Inefficient resource utilization may result in bottlenecks and performance degradation, particularly on systems with limited resources.

Proposed System:

While the existing single-threaded port scanner effectively identifies open ports, its limitations in terms of performance, resource utilization, scalability, and user experience highlight the need for our proposed solution. Our port scanner addresses these drawbacks by implementing concurrent scanning techniques, efficient resource management, and real-time feedback mechanisms to enhance performance, scalability, and user satisfaction.

4.1.2 Proposed System Architecture:

Introduction to Proposed System Architecture:

The proposed system architecture for the port scanner aims to overcome the limitations of traditional single-threaded scanners, including performance bottlenecks and scalability issues. By incorporating concurrent scanning techniques, efficient resource management, and real-time feedback mechanisms, the proposed architecture strives to enhance scanning performance, scalability, and user experience. Key objectives include expediting the scanning process, seamlessly handling larger networks, and providing real-time updates and

customizable parameters for improved user satisfaction. Overall, the proposed architecture seeks to meet the evolving needs of network security professionals by elevating the efficiency, scalability, and usability of port scanning processes.

Overview of Proposed Components:

Input Handling:

The proposed system architecture for the port scanner encompasses key components to optimize the scanning process and enhance user interaction. Input handling ensures accurate parameter validation and parsing. Thread management efficiently allocates and synchronizes multiple threads for concurrent port scanning, optimizing resource utilization. Concurrent scanning techniques expedite the probing of target ports while minimizing resource consumption. Result aggregation consolidates findings for comprehensive analysis, while real-time feedback provides instant updates on open ports, improving monitoring capabilities. Overall, this architecture delivers a robust and efficient port scanning solution, mitigating the shortcomings of traditional single-threaded scanners.

Thread Management:

Scanning Process:

In the concurrent scanning process, each thread independently attempts to establish connections with the target host by iterating through a specified range of ports. It creates sockets for each port and initiates connection attempts, recording successful connections as open ports and failures as closed or filtered ports. To enhance efficiency and minimize resource consumption, optimizations such as setting appropriate timeouts, randomizing port scanning order, and utilizing non-blocking I/O operations are employed. These strategies improve scanning speed, distribute the scanning load evenly, and ensure robustness while conserving resources.

Connection Establishment:

In port scanning, the scanner sequentially selects ports within a specified range and sends connection requests to the target host's IP address. If a response is received within a set timeout period, the port is marked as open; otherwise, it's considered closed or filtered. To ensure robustness, timeouts prevent indefinite waits, while retries may be employed for reliability in case of network issues. This systematic approach ensures accurate and

comprehensive scanning results by efficiently probing target ports while managing potential connection challenges.

Result Aggregation:

Results from individual threads are aggregated and stored centrally through a systematic process in the port scanning architecture. Typically, data structures like lists or dictionaries are employed to collect and organize scan results. Each thread updates the central data structure with its findings, allowing for centralized storage and analysis of port statuses. Algorithms may include simple insertion or merging techniques to combine results efficiently, ensuring that the aggregated data accurately represents the scan outcomes. This centralized aggregation facilitates comprehensive analysis and reporting of port scanning results.

Real-Time Feedback:

The optional verbose output feature in the port scanning tool offers real-time updates on open ports detected during the scanning process. This feature enhances user visibility by providing immediate feedback on the progress of the scan, allowing users to monitor the scan's status in real-time. Users can observe the incremental discovery of open ports as they occur, enabling them to assess the effectiveness and efficiency of the scanning process. By offering real-time feedback, this feature empowers users to make informed decisions and adjustments during the scan, ultimately improving their ability to manage and interpret scanning results effectively.

Output Display:

Upon completion of the scanning process, aggregated results are typically formatted and displayed to the user in a clear and organized manner. This may involve presenting a summary of open, closed, and filtered ports, along with their corresponding port numbers. Additionally, the total number of open ports found and the time taken for the scan may be provided to offer valuable insights into the network's security posture and the efficiency of the scanning operation. Such additional information or statistics aid users in assessing the scan's outcomes comprehensively and making informed decisions regarding network security measures.

-.

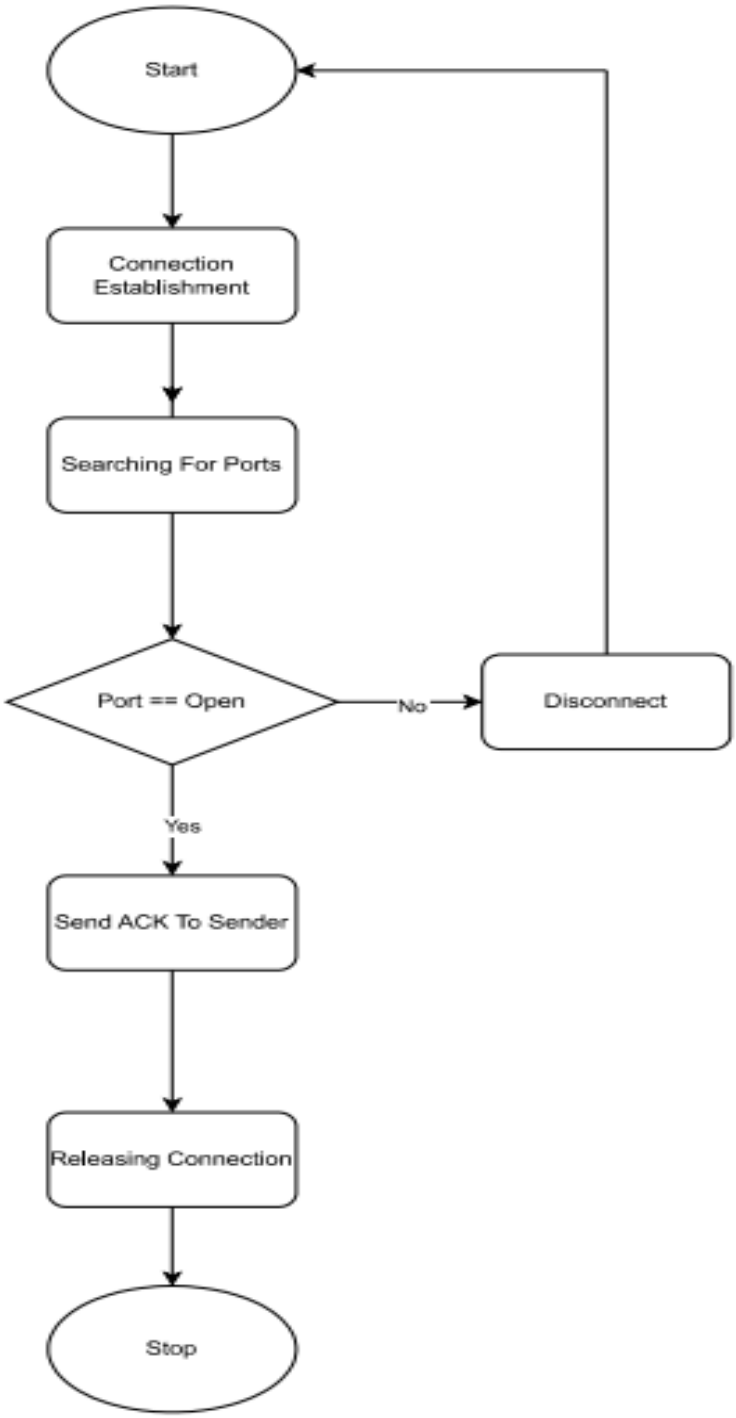

Fig 4.2 Flow diagram

Experiment No. 8

Chapter 5

Project Design & Process Workflow

This chapter consist of Process design and workflow for a port scanner involves defining the methodology and steps to efficiently scan network ports for vulnerabilities. It encompasses requirement analysis, selection of scanning techniques, tool configuration, validation through testing, and documentation. The process workflow includes initialization, scan execution, result analysis, reporting, and response/mitigation. By following a systematic approach, organizations can enhance their network security posture by identifying and addressing potential security risks effectively.

5.1 Use Case Diagram / Activity Diagram /DFD

Use Case Diagram:

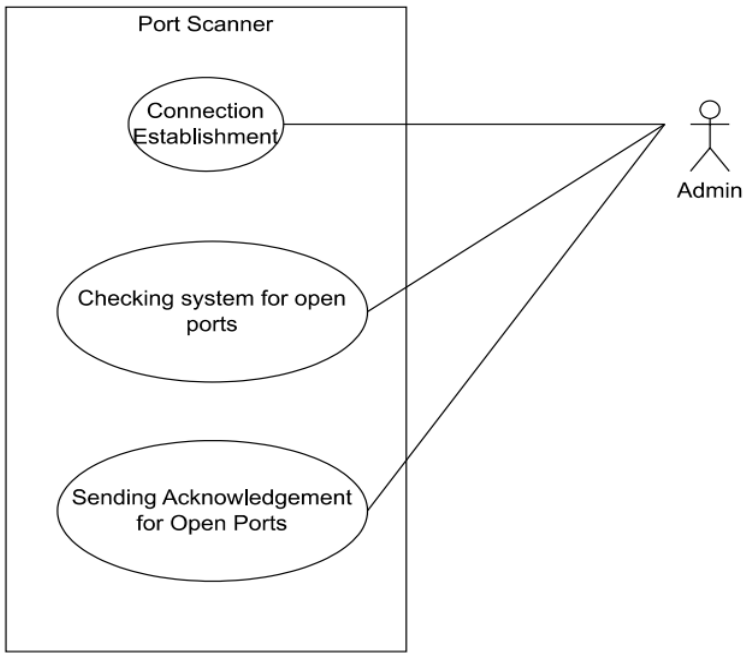

Fig 5.1 Use case of Port Scanner

Activity Diagram:

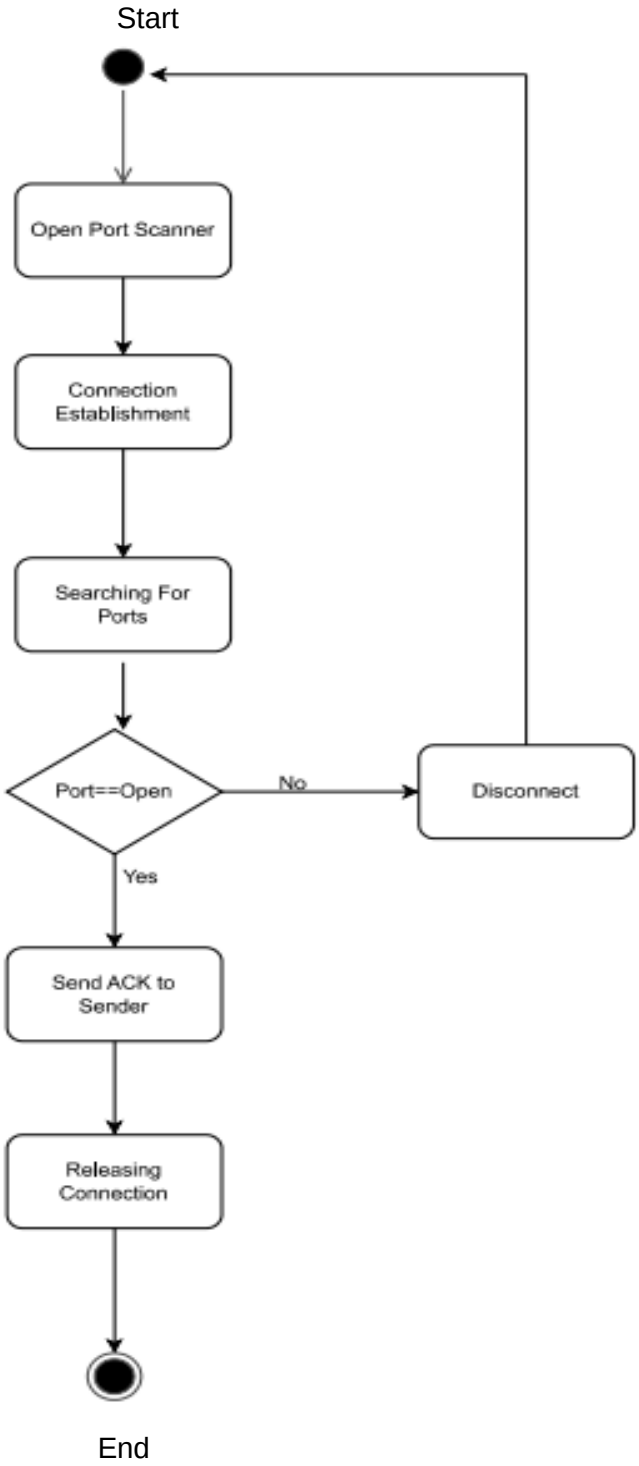

Fig.5.2 Activity Diagram of Port scanner

DFD :

DFD Level 0

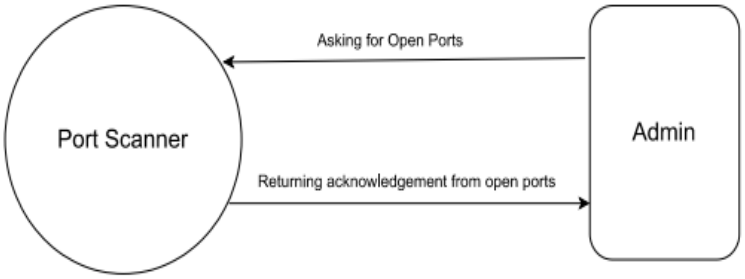

Fig 5.3 DFD of Port scanner

5.2 Algorithm:

START

Step 1: Initialize an empty list open_ports to store the results of open ports.

Step 2: Define the prepare_args function to parse command-line arguments for the script.

Step 3: Define the prepare_port generator function to yield a sequence of ports to scan.

Step 4: Define the scan_port function to attempt connections to each port and record open ones.

Step 5: Define the prepare_threads function to create and manage multiple threads for scanning.

- In the main execution block:

Step 6: Parse the arguments and prepare the port range.

Step 7: Record the start time of the scan.

Step 8: Initiate the threads to start scanning.

Step 9: Wait for all threads to complete.

Step 10: Record the end time of the scan.

Step 11: If verbose mode is enabled, print the list of open ports and the total time taken.

END

Overall, the script follows these steps:

- Parse command-line arguments.
- Generate a sequence of ports to scan.
- Manage multiple threads for scanning.
- Perform port scanning using TCP connections.
- Output the list of open ports and the time taken for scanning.

5.3 Hardware and Software Specifications:

Table 5.1 Hardware & Software specification

Sr. no.	**Component**	**Specification**
	Hardware	
1.	Processor	Intel core i5 or equivalent
2.	RAM	8GB
3.	Network interface	Ethernet(10/100/1000Mbps)
4.	Storage	SSD
	Software	**Specification**
1.	Operating system	Cross-platform(Windows,MacOS,linux)
2.	Python Interpreter	Python 3.x
3.	Required packages	Argparse,socket,Threading
4.	Development IDE/ Tool	Any text editor(e.g., VS code,Sublime

This table outlines the suggested hardware specifications, including processor, memory, and network interface, as well as the required software specifications, such as the operating system, Python interpreter version, necessary Python packages, and a recommended development environment

Experiment 9

Chapter 6

Result and Application

This chapter consist of results and applications of port scanners that involve identifying open ports and potential vulnerabilities within network infrastructure. These tools provide insights into network security posture, aiding in threat detection, risk assessment, and penetration testing. With their ability to uncover entry points for unauthorized access or malicious activity, port scanners play a crucial role in strengthening cybersecurity defenses. Their applications range from routine network maintenance to proactive security measures, helping organizations safeguard sensitive data and infrastructure from cyber threats.

6.1 Sample of Inputs, Outputs and GUI Screenshots

Input:

```
open_ports = []
def prepare_args():
 parser = ArgumentParser(description="python based fast port scannner",
usage="%(prog)s 192.168.1.2",
 epilog="Example - %(prog)s -s 20 -e 40000 -t 500 -V 192.168.1.2")
 parser.add_argument(metavar="IPv4", dest="ip", help="host to scan")
 parser.add_argument("-s", "--start", dest="start", metavar="", type=int,
help="staring port", default=1)
 parser.add_argument("-e", "--end", dest="end", metavar="", type=int,
help="ending port", default=65535)
 parser.add_argument("-t", "--threads", dest="threads", metavar="", type=int,
help="threads to use ", default=500)
 parser.add_argument("-V", "--verbose", dest="verbose", action="store_true",
help="verbose output")
 parser.add_argument("-v", "--version", action="version", version="%(prog)s 1.0",
help="display version")
 args = parser.parse_args()
 return args
def prepare_port(start: int, end: int):
 for port in range(start, end + 1):
 yield port
def scan_port():
```

```
 while True:
 try:
 s = socket.socket()
 s.settimeout(1)
 port = next(ports)
 s.connect((arguments.ip, port))
 open_ports.append(port)
 if arguments.verbose:
 print(f"\r{open_ports}", end="")
 except (ConnectionRefusedError, socket.timeout):
 continue
 except StopIteration:
 break
def prepare_threads(threads: int):
 thread_list = []
 for _ in range(threads + 1):
 thread_list.append(Thread(target=scan_port))
 for thread in thread_list:
 thread.start()
 for thread in thread_list:
 thread.join()
 if __name__ == "__main__":
 arguments = prepare_args()
 ports = prepare_port(arguments.start, arguments.end)
 start_time = time()
 prepare_threads(arguments.threads)
 end_time = time()
 if arguments.verbose:
 print()
 print(f"open ports found - {open_ports}")
 print(f"time taken - {round(end_time - start_time, 2)}")
```

Fig.6.1 Input of port scanner

Output :

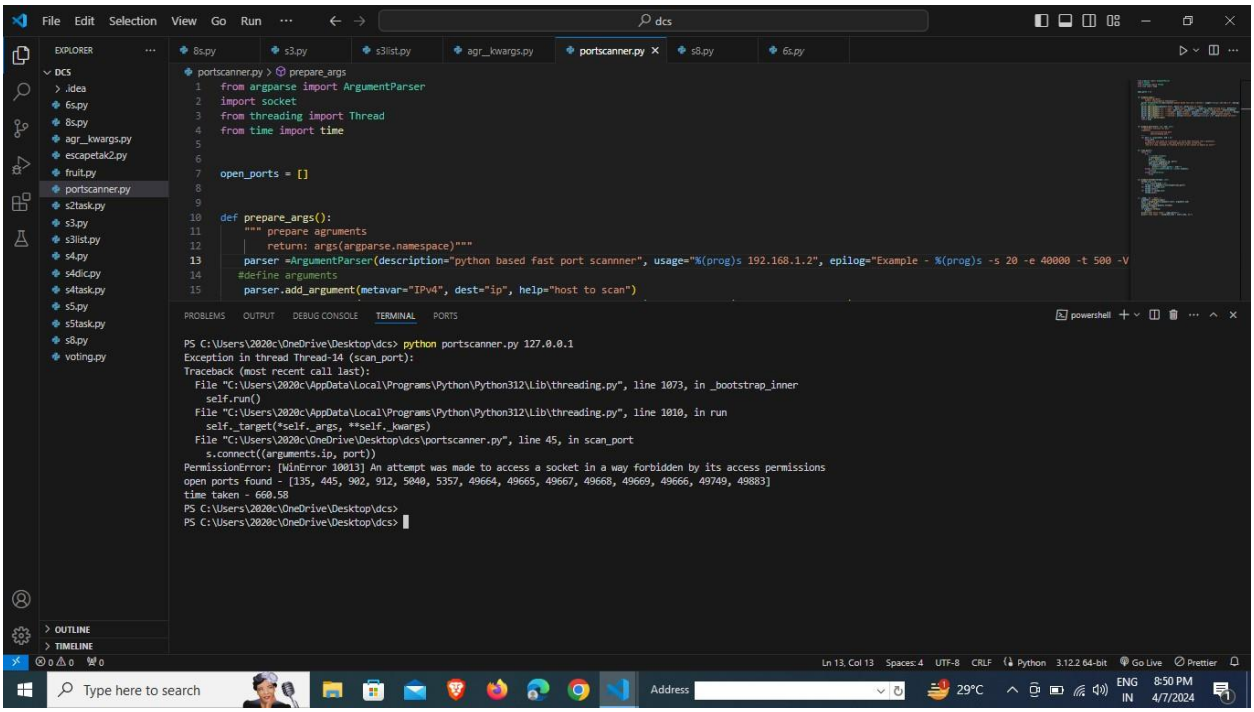

Fig. 6.2 Output of Portscanner

6.2 Evaluation Parameters:

- **Performance:** Measure the speed and efficiency of the port scanner in scanning a range of ports on target hosts. Evaluate factors such as scan time, throughput, and resource utilization.
- **Accuracy:** Assess the accuracy of port scanning results by comparing identified open ports with ground truth data. Evaluate false positives/negatives and the ability to correctly identify port states (open, closed, filtered).
- **Scalability:** Determine the scalability of the port scanner by analyzing its performance as the size of the target network or port range increases. Evaluate whether the tool can handle large-scale scanning operations efficiently.

- **Ease of Use:** Evaluate the usability and user-friendliness of the port scanner, including the clarity of documentation, ease of installation, configuration options, and the intuitiveness of the user interface.

- **Flexibility:** Assess the flexibility of the port scanner in terms of customization and extensibility. Evaluate the ability to modify scanning parameters, integrate with other tools or frameworks, and extend functionality through plugins or scripting.
- **Stealthiness:** Measure the stealthiness of the port scanner by analyzing its ability to conduct scans without triggering intrusion detection systems or raising alarms. Evaluate techniques used to minimize detection and avoid being labeled as suspicious activity.
- **Reliability:** Determine the reliability of the port scanner by assessing its stability, robustness, and error handling capabilities. Evaluate how the tool handles unexpected network conditions, timeouts, or errors encountered during scanning.
- **Community Support:** Consider the availability of community support, active development, and maintenance of the port scanner. Evaluate the responsiveness of developers to bug reports, feature requests, and the availability of online forums or documentation resources.

6.3 Performance Evaluation (Tables)

Table 6.1 performance Evaluation of Different port scanner

Parameter	Advanced Port scan	Nmap	Angry IP scan	Unicorn scan
Performance	High speed scanning capabilities	Capable of scanning large network quickly	Suitable for scanning smaller network or indivisual hosts	Offers fast scanning, especially for certain protocols, but may vary depending on network size and configuration.
Accuracy	Accuracy may vary depending on the methodology used and implementation quality.	Renowned for its accuracy, provides detailed information about discovered hosts and ports.	Generally accurate in identifying active hosts and open ports, but occasional discrepancies may occur.	Known for accurate results, providing comprehensive information about discovered services and vulnerabilities.
Scalability	Scalability depends on the efficiency of the implementation and resource utilization.	Highly scalable, capable of handling large-scale scanning operations with ease.	Suitable for scanning smaller networks or individual hosts, may encounter limitations with larger networks.	Offers scalability for various network sizes, but may require optimization for very large networks
Ease of Use	Ease of use varies depending on the specific implementation and user interface design.	Offers a wide range of features with various complexity levels, may require some learning curve for novice users.	Known for its simple and intuitive interface, suitable for users of all skill levels.	Provides a user-friendly interface, with straightforward options for conducting scans and analyzing results.

Experiment no. 10

References and bibliography

REFERENCES:

1) Zhang, Xu, Jeffrey Knockel, and Jedidiah R. Crandall. "Original SYN: Finding machines hidden behind firewalls." *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015.
2) Singh, Rajni Ranjan, and Deepak Singh Tomar. "Network forensics: detection and analysis of stealth port scanning attack." *International Journal of Computer Networks and Communications Security* 3.2 (2015): 33-42.
3) Zhao, Jinxiong, et al. "Research on the Speed and Accuracy of Full Port Scanning." *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 6. IEEE, 2023.
4) Boyanov, Petar. "IMPLEMENTATION OF MODIFIED NETWORK PORT SCANNER FOR WINDOWS BASED OPERATING SYSTEMS: IMPLEMENTATION OF MODIFIED NETWORK PORT SCANNER FOR WINDOWS BASED OPERATING SYSTEMS." *Journal scientific and applied research* 23.1 (2022): 73-84.
5) Wang, Yien, and Jianhua Yang. "Ethical hacking and network defense: choose your best network vulnerability scanning tool." *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2017.
6) Lyon, Gordon. "Nmap network mapper." *Recuperado de https://nmap. org* (2019).
7) Ferguson, Bernard, Anne Tall, and Denise Olsen. "National cyber range overview." *2014 IEEE Military communications conference*. IEEE, 2014.
8) Workman, Russell. *TCP over military networks*. Swansea University (United Kingdom), 2004.
9) Wright, Gary R., and W. Richard Stevens. *TCP/IP illustrated, volume 2: The implementation*. Addison-Wesley Professional, 1995.
10) Comer, Douglas E. *Internetworking con TCP/IP*. Vol. 1. Pearson Italia Spa, 2006.
11) Comer, Douglas E. *Internetworking with TCP/IP*. Addison-Wesley Professional, 2013.
12) Maimon, Uriel. "Port Scanning without the SYN flag." *Phrack Magazine* 7.49 (1996): 49-15.
13) De Vivo, Marco, et al. "A review of port scanning techniques." *ACM SIGCOMM Computer Communication Review* 29.2 (1999): 41-48.
14) LACORE, UCV. "A Review of Port Scanning Techniques."
15) De Vivo, Marco, et al. "A review of port scanning techniques." *ACM SIGCOMM Computer Communication Review* 29.2 (1999): 41-48.
16) Comer, Douglas E., and John C. Lin. "Probing TCP implementations." *Usenix Summer*. 1994.
17) Vugrin, Eric D., et al. "Cyber threat modeling and validation: port scanning and detection." *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*. 2020.
18) Ring, Markus, Dieter Landes, and Andreas Hotho. "Detection of slow port scans in flow-based network traffic." *PloS one* 13.9 (2018): e0204507.
19) Upadhya, Abhinav, and B. K. Srinivas. "A Survey on different Port Scanning Methods and the Tools used to perform them." *international journal for research in applied science and engineering technology* 8.5 (2020).