International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Advanced Port Scanner Tool Using Python

Pushkar Chaudhari[1], Vaibhav Thakre[2], Tushar Chaudhari[3], Tanya Bhaute[4], Dr. Rais Khan[5]
Sandip University, Nashik, India

Abstract- Port scanning is a fundamental technique used in cybersecurity for identifying active services and potential vulnerabilities in networked systems. As modern networks grow in complexity, efficient and scalable scanning tools become increasingly important for administrators and security researchers. This paper presents the design and development of an advanced multithreaded port scanner implemented in Python and executed on the Linux operating system. The proposed system aims to provide efficient port discovery, faster scanning performance, and structured reporting for network security analysis. Unlike traditional sequential scanners, the proposed approach utilizes parallel execution techniques to analyze multiple ports simultaneously. The architecture includes modules for user input handling, scanning engine management, multithreading coordination, result processing, and reporting. Experimental evaluation demonstrates improved scanning speed and reliability compared to conventional scanning approaches.

Keywords – Network Security, Port Scanning, Python, Linux, Multithreading, Cybersecurity, TCP Scanning.

I. INTRODUCTION

Computer networks are an essential part of modern information systems, supporting communication, data exchange, and distributed computing. However, the increasing reliance on networked systems also introduces significant security risks. Attackers often exploit open ports and vulnerable services to gain unauthorized access to systems.

Port scanning is widely used in penetration testing, vulnerability analysis, and network administration. Each network service communicates through a specific port number, allowing applications such as web servers, mail servers, and file transfer services to exchange information across networks.

II. RELATED WORK

Network scanning techniques have evolved significantly over the past two decades. Early scanners used sequential scanning mechanisms that tested one port at a time. Modern scanners incorporate parallel processing techniques and packet analysis methods.

Python-based security tools are widely used because they provide rapid development capabilities and strong networking libraries. Concurrent programming techniques allow scanners to analyze multiple ports simultaneously, significantly improving performance.

III. SYSTEM ARCHITECTURE

The proposed port scanning system follows a modular architecture that separates different operational responsibilities into individual components. This design improves scalability, maintainability, and performance.

Modules include:

- User Input Module
- Port Scanning Engine
- Multithreading Controller
- Result Processing Unit
- Output Reporting Module

IV. METHODOLOGY

The scanning process begins when the user provides a target host and port range. If a domain name is provided, it is resolved into an IP address.

Multiple threads are initialized to scan ports concurrently. Each thread attempts to establish a connection with a designated port on the target system. Successful connections indicate open ports while failed attempts indicate closed ports.

Once scanning completes, the system aggregates the results and generates a structured report.

V. EXPERIMENTAL EVALUATION

Testing was performed in a Linux environment to evaluate scanning performance. The multithreaded implementation

demonstrated faster scanning performance compared with traditional sequential scanning approaches.

Common services such as SSH, HTTP, and HTTPS were successfully detected during experiments, demonstrating the reliability of the system.

VI. ADVANTAGES

- Faster scanning speed using multithreading
- Lightweight and easy to deploy
- Flexible architecture for future enhancements
- Useful for cybersecurity education and research

VII. LIMITATIONS AND FUTURE WORK

The current system focuses primarily on TCP port detection. Future enhancements may include UDP scanning, stealth scanning, vulnerability detection, and graphical user interfaces.

VIII. CONCLUSION

This paper presented the design and development of an advanced port scanner using Python on Linux. The system demonstrates how concurrent programming can significantly improve scanning performance while maintaining accuracy in detecting open ports.

REFERENCES

1. Lyon, G. Nmap Network Scanning.
2. Stallings, W. Network Security Essentials.
3. Python Software Foundation Documentation.
4. RFC 793 Transmission Control Protocol.