Assignment Report – Mini Port Scanner

Course: Ethical Hacking

Name: Adi Sankar

Roll Number: 2460308

Index:

1. Introduction
2. Objective
3. Tools & Technologies Used
4. Implementation Steps
5. Problems Faced
6. Output Examples
7. Educational Benefits
8. Conclusion

Introduction:

The Mini Port Scanner is a lightweight network security tool that scans a target system for open ports. This project demonstrates fundamental concepts in networking and cybersecurity, such as TCP/UDP communication, socket programming, and reconnaissance techniques.

Objective:

- To identify open and closed ports on a target host.
- To understand basic network scanning techniques.
- To create a simple yet effective port scanning tool using programming.

Tools & Technologies Used:

- To identify open and closed ports on a target host.
- To understand basic network scanning techniques.
- To create a simple yet effective port scanning tool using programming.

Implementation Steps:

Step 1:

Install Python (if not installed)

Step 2:

Install Required Library

pip install socket

Step 3: Write the Python CLI Mini Port Scanner

```
#!/bin/bash

echo "================================"
echo "     MINI PORT SCANNER"
echo "================================"

# Ask for target IP or domain
read -p "Enter the IP or domain to scan: " TARGET

# Get current date
DATE=$(date +%F)
```

 # Output file name with date

OUTPUT="scan_${DATE}.log"

echo "Scanning top 1000 ports on $TARGET..."

echo "Saving results to $OUTPUT"

 # Scan using nmap

nmap -T4 --top-ports 1000 "$TARGET" -oN "$OUTPUT"

echo "Scan complete!"

echo "Results saved in $OUTPUT"

Step 4: Run the Script

python mini_port_scanner.py

Step 5: Example Output

The script uses the `nmap` tool, which is a powerful port scanner used in cybersecurity.

- It takes an IP address or domain name from the user (e.g., 192.168.1.1 or scanme.nmap.org).

- It runs: nmap -T4 --top-ports 1000 <target>

- -T4 = faster scan timing

- --top-ports 1000 = scans the 1000 most commonly used TCP ports

- The results are saved to a log file named like scan_<date>.log

Problem Faced:

• **Permission Restrictions**: Some ports require administrative privileges to scan.

• **Network Latency**: Slow response times for certain hosts.

• **Firewall Restrictions**: Some ports may be filtered or blocked.

Output Examples:

Each line in the result shows:
- Port Number: e.g., 22, 80, 443
- State: Whether the port is:
  - open → a service is actively listening
  - closed → no service is listening
  - filtered → blocked by firewall
- Service: The known service usually running on that port (e.g., SSH, HTTP)

Example:
22/tcp open ssh
80/tcp open http

This means the target is running:
- SSH (Secure Shell) on port 22 → used for secure remote login
- HTTP (Web server) on port 80 → standard web traffic

Educational Benefits:

- Hands-on experience with **socket programming**.
- Understanding **network security basics**.
- Familiarity with **network protocols** (TCP, UDP).
- Exposure to **cybersecurity tools** and penetration testing basics

Conclusion:

The Mini Port Scanner project successfully identifies open ports on a target system, demonstrating the practical application of network scanning fundamentals.

It bridges the gap between theoretical networking concepts and real-world cybersecurity skills, enhancing the understanding of TCP connections, socket programming, and the services that operate on various ports. By revealing potential vulnerabilities through open ports, the project emphasizes the importance of proper firewall configuration and system hardening. Implementing the scanner in Python provides a lightweight, cross-platform, and beginner-friendly approach that remains easily customizable for future development.

This project also serves as a stepping stone to more advanced penetration testing tools such as Nmap, while fostering ethical hacking practices by reinforcing the need for scanning only with proper authorization.

The process of building and testing the scanner strengthened problem-solving abilities, particularly in addressing network-related challenges, and laid the groundwork for enhancements

such as multi-threading for faster scans and the inclusion of UDP scanning capabilities..