PROJECT-4 PORT SCANNER USING PYTHON

Objective:

To create a Python-based port scanner that:

- Accepts a target IP or domain
- Scans multiple ports
- Identifies and lists open ports
- Uses multithreading for faster scanning

Requirements:

Component	Description
Python	Programming language
socket module	To connect to target ports
threading module	To perform parallel scans
Command-line interface	To input host and display results

Technologies Used

- **Language:** Python 3.x
- **Modules:** socket, threading
- **Platform:** Cross-platform (Windows)

Create a new file(eg:portscan.py) and add python code in it and run it in command prompt.

Python code:

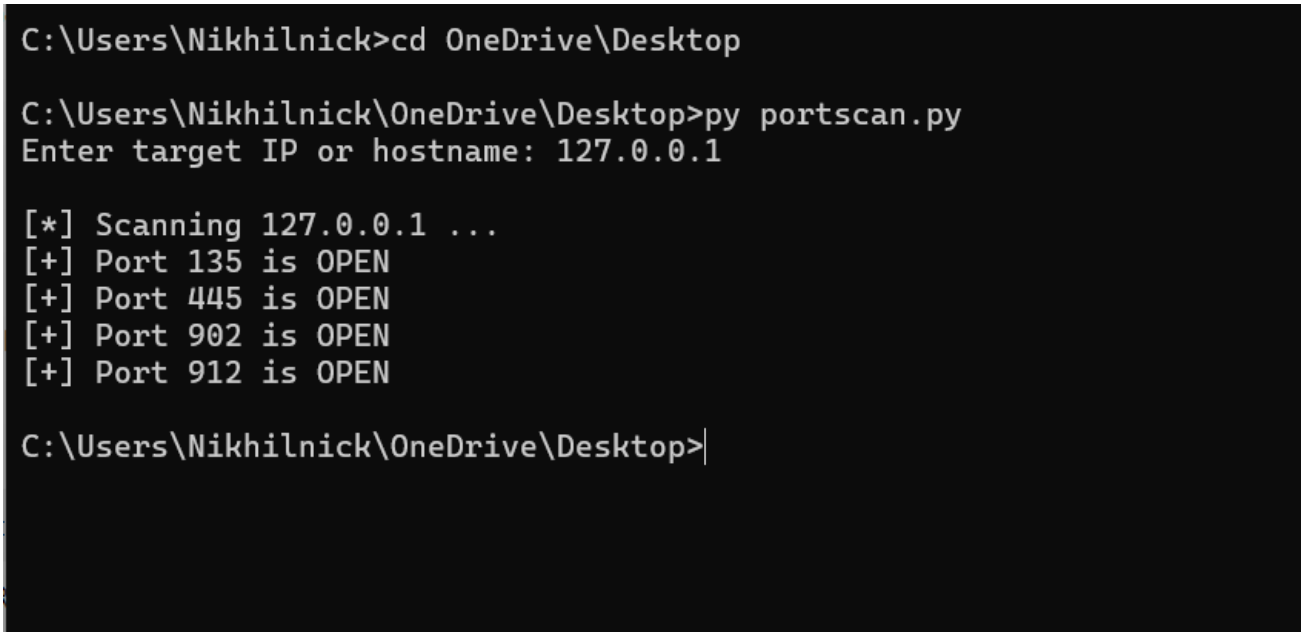
```
import socket
import threading

# Lock for thread-safe print
print_lock = threading.Lock()

# Function to scan a single port
def scan_port(host, port):
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.settimeout(1)  # Timeout for connection attempt
        result = s.connect_ex((host, port))  # Returns 0 if port is open
        if result == 0:
            with print_lock:
                print(f"[+] Port {port} is OPEN")
        s.close()
    except Exception:
        pass

# Function to scan multiple ports
def run_scanner(host, ports):
    print(f"\n[*] Scanning {host} ...")
    for port in ports:
        thread = threading.Thread(target=scan_port, args=(host, port))
        thread.start()
```

```
# Entry point
if __name__ == "__main__":
    target = input("Enter target IP or hostname: ")
    try:
        target_ip = socket.gethostbyname(target)
        ports_to_scan = range(1, 1025)  # Common ports
        run_scanner(target_ip, ports_to_scan)
    except socket.gaierror:
        print("[!] Invalid hostname or IP address.")
```



```
C:\Users\Nikhilnick\OneDrive\Desktop>py portscan.py
Enter target IP or hostname: 192.168.25.1

[*] Scanning 192.168.25.1 ...
[+] Port 135 is OPEN
[+] Port 139 is OPEN
[+] Port 445 is OPEN
[+] Port 902 is OPEN
[+] Port 912 is OPEN

C:\Users\Nikhilnick\OneDrive\Desktop>
```

```
C:\Users\Nikhilnick\OneDrive\Desktop>py portscan.py
Enter target IP or hostname: 65.2.45.171

[*] Scanning 65.2.45.171 ...
[+] Port 80 is OPEN

C:\Users\Nikhilnick\OneDrive\Desktop>
```

```
C:\Users\Nikhilnick\OneDrive\Desktop>py portscan.py
Enter target IP or hostname: 177.67.133.169

[*] Scanning 177.67.133.169 ...
[+] Port 33 is OPEN
[+] Port 90 is OPEN

C:\Users\Nikhilnick\OneDrive\Desktop>
```

Code Description

Input

- User enters a **target hostname** or **IP address**

Processing

- The script:
    1. Converts hostname to IP (if needed)
    2. Iterates over a list of ports (1 to 1024)
    3. Tries connecting to each port
    4. Logs ports that respond as **open**
    5. Uses **threads** to speed up scanning

Conclusion:

This project successfully implements a **multi-threaded port scanner** using Python to identify open ports on a target system. By using socket and threading, it efficiently scans ports and helps in assessing basic network security. It provides foundational knowledge for understanding how attackers identify exposed services, and how defenders can mitigate risks by closing or securing those ports.