Python Socket Port Scanner

Python Socket Port Scanner

# Introduction:

A port scanner is a network utility used to identify open ports on a system. It helps network administrators and cybersecurity professionals detect potential vulnerabilities or unauthorized services. Python, with its built-in socket library, provides an effective and simple way to implement a port scanner.

# Working Principle:

The scanner works by attempting to establish a connection to a list of ports on a given IP address. If the connection is successful, it means the port is open and a service is listening on it. If the connection fails, the port is likely closed or filtered by a firewall.

# Key Components of the Code:

- socket.socket() creates a new socket object.

- connect\_ex((host, port)) attempts to connect to a specific port. It returns 0 if the connection is successful.

- A loop is used to scan a range of ports (usually 1 to 1024 for quick scanning).

- Optional: threading can be added to speed up the scanning process.

# Sample Code:

import socket

target = input("Enter target IP: ")

# for port in range(1, 1025):

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.settimeout(0.5)

result = s.connect\_ex((target, port))

# if result == 0:

print(f"Port {port} is open")

s.close()

# Applications:

- Network security auditing

- Troubleshooting network services

- Penetration testing

# Limitations:

- Slow for large port ranges without threading

- May be blocked or flagged by intrusion detection systems (IDS)

- Cannot bypass firewalls or advanced security filters

# Conclusion:

Python socket port scanners are simple yet powerful tools for checking open ports. While basic, they are useful for small-scale testing and learning about network security. For more advanced scanning, tools like Nmap offer greater features and flexibility.

Assignment 19: Python Socket Port Scanner

Create a script that scans ports 1–100 on a given domain.  
Limit: 15 lines  
Add sleep between scans and proper output formatting.

Code:

import socket  
import time  
  
target = input("Enter target domain: ")  
ip = socket.gethostbyname(target)  
print(f"\nScanning {target} ({ip})...\n")  
  
for port in range(1, 101):  
 s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 s.settimeout(0.5)  
 result = s.connect\_ex((ip, port))  
 if result == 0:  
 print(f"[+] Port {port:>3} is OPEN")  
 else:  
 print(f"[-] Port {port:>3} is closed")  
 s.close()  
 time.sleep(0.1)