**HONOURS ASSIGNMENT – ETHICAL HACKING (CSHO331CSP)**

**Assignment 19: Python Socket Port Scanner**

**Methodology:**

Write a 15-line Python script to scan ports 1–100 on a given domain, use sockets, include delays, format output, and understand the "how and why."

Essential features needed:

Accept a **domain/IP input**.

**Resolve hostname** to IP.

**Scan TCP ports** from 1 to 100.

Use **sockets** (not external libraries).

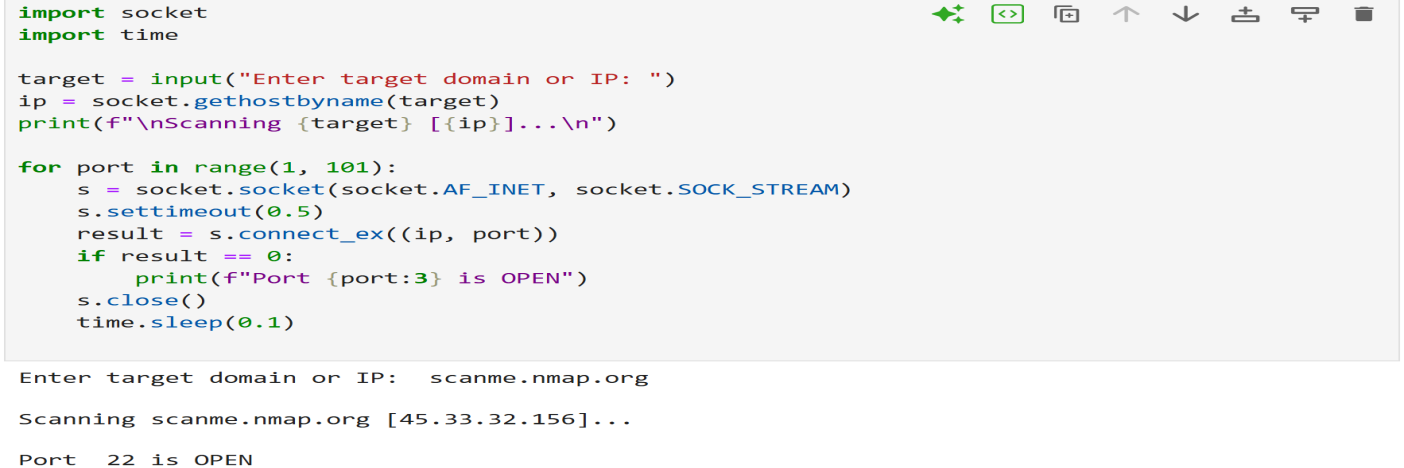
Include **timeout** and **sleep** between scans.

Print only **open ports**, formatted cleanly.

Compared the script’s capabilities with professional tools like nmap to highlight:

What can we do with raw sockets? What advanced tools do beyond this? (like parallel scans, stealth modes, UDP, etc.)

**Code and screenshot:**



**Findings:**

* Port Scanning Is Fundamentally Simple - Even advanced scanning tools are built on simple low-level operations the complexity comes from scale, optimization, and evasion features.
* Open Ports doesn’t mean Safe Ports - Port scanning is just the first step security assessments must go deeper such as service analysis, version detection, vulnerability scans.
* Delays Help Prevent Detection - Even basic port scanning has ethical and legal implications responsible scanning practices are important.
* Sockets Give You Fine-Grained Control - Using sockets directly is a great way to learn how TCP handshakes and port availability work.
* Only TCP Is Scanned - Real scanners like nmap handle multiple protocols, service detection, and fingerprinting our basic script doesn’t.

**Conclusions:**

* Port Scanning Can Be Built from Scratch
* TCP/IP Behavior Becomes Tangible
* Timeouts and Delays Are Critical
* Formatted Output Improves Clarity
* Security Awareness Grows