Port Scanner Using Python

A technical implementation focused on identifying open TCP ports on target systems, addressing the risk of exposed services running on unknown ports, and applying core cybersecurity concepts related to service discovery and attack surface analysis.

Author : Akash Motghare

Core Scanning Approach & Design

TCP Socket-Based Scanning

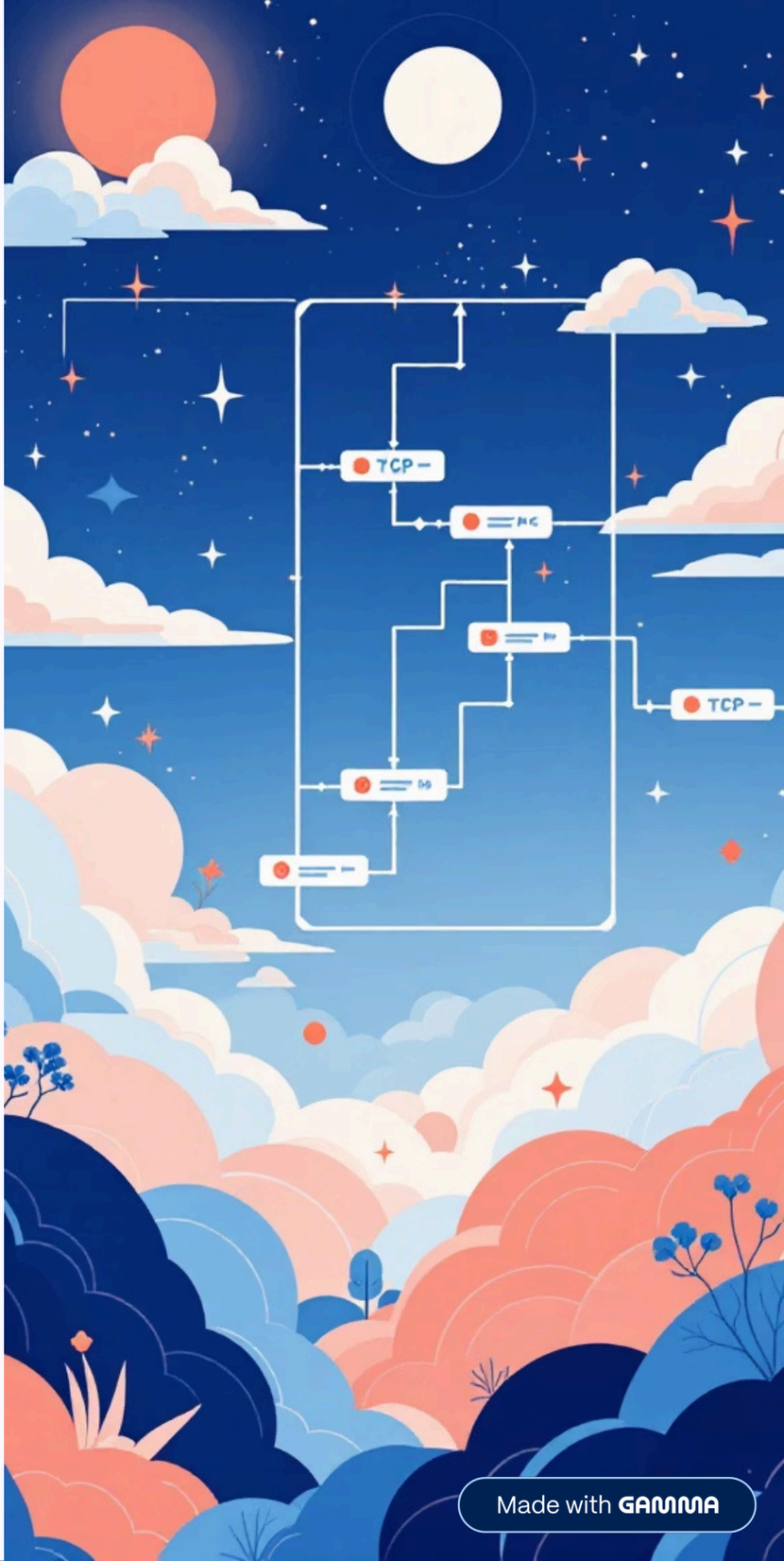
Implements low-level socket programming to establish connections with target ports and evaluate their availability.

Common Port Targeting

Focuses scanning efforts on frequently used ports associated with standard network services and protocols.

Handshake Detection

Identifies open ports by detecting successful TCP three-way handshake completion between scanner and target.



Key Results from Implementation & Testing

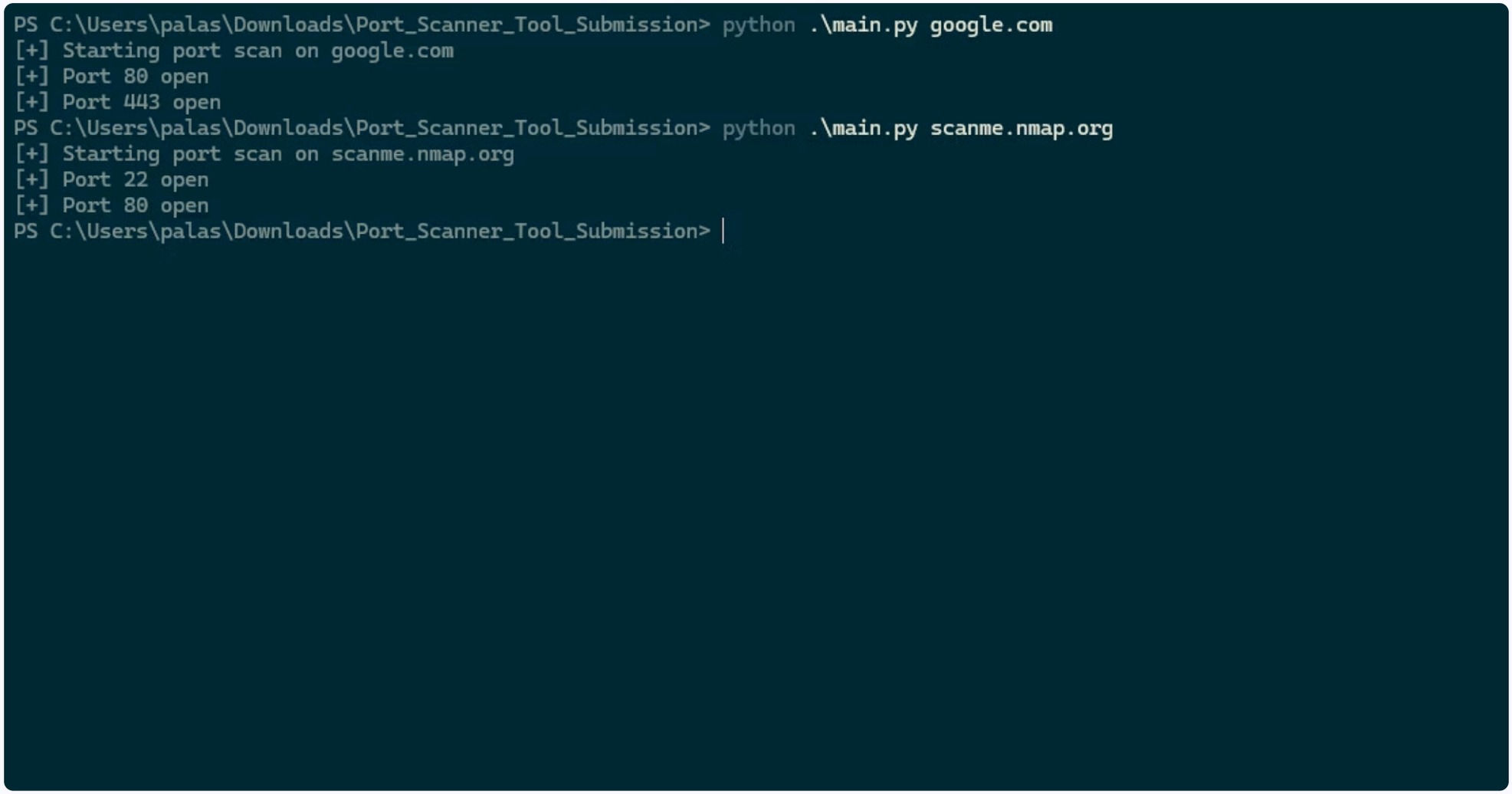
Port Discovery

- Successfully identified open ports on target hosts
- Validated connection establishment across multiple test environments
- Confirmed accurate detection of listening services

Service Detection

- Detected commonly exposed services including HTTP and SSH
- Mapped ports to their associated service protocols
- Demonstrated how open ports reveal active services on systems

Project Screenshot



Next Steps & Ownership

01

Full Port Range Support

Extend scanner capabilities to support scanning across the complete port range of 1-65535 for comprehensive coverage.

02

Multithreading Implementation

Integrate concurrent scanning techniques to significantly improve scanning speed and reduce overall execution time.

03

Enhanced Reporting

Develop detailed logging and reporting mechanisms to document scan results and facilitate analysis.

Project Ownership: Akash Motghare – Responsible for design, implementation, testing, and documentation of the port scanning tool.