**Project Report for Cybersecurity Mini Project**

**Title: GUI-Based Nmap Application for Scanning Basic TCP Applications**

**Abstract:** This project aims to develop a Graphical User Interface (GUI)-based application for network scanning using the Nmap tool. The application allows users to enter a target IP address or hostname and scan for open TCP ports in the range 1-1024. The project leverages Python, Tkinter for the GUI, and the Nmap library for executing network scans. It provides an intuitive interface for users to analyze network security and detect open ports.

**Introduction:** Network security is a crucial aspect of cybersecurity, requiring constant monitoring for vulnerabilities. Nmap (Network Mapper) is a powerful open-source tool used for network discovery and security auditing. In this project, we create a Python-based GUI that integrates with Nmap to scan basic TCP applications. Users can input a target address and view scan results in a user-friendly interface.

**Objectives:**

* Develop a GUI-based application for scanning open TCP ports using Nmap.
* Provide an easy-to-use interface for entering a target IP or hostname.
* Display scan results in a structured manner.
* Handle errors related to invalid inputs or missing dependencies.

**Methodology:**

1. **Tools and Technologies:**
   * Programming Language: Python
   * Libraries Used: Tkinter (for GUI), Nmap (for network scanning), OS (for system checks)
2. **Scanning Process:**
   * The user enters a target IP address or hostname.
   * The application checks for the Nmap installation path.
   * Nmap executes a scan on ports 1-1024 using the '-T4' timing option.
   * The results are displayed in the GUI output window.
3. **Error Handling:**
   * Ensures Nmap is installed and accessible.
   * Handles invalid IP addresses or hostnames.
   * Catches exceptions during scanning to prevent application crashes.

**Implementation Details:**

* **GUI Design:**
  + Entry field for target IP or hostname.
  + Button to initiate the scan.
  + Text output box for displaying scan results.

**Code Implementation:**

import tkinter as tk

from tkinter import messagebox

import nmap

import os

# Set the correct path for Nmap

NMAP\_PATH = "C:\\Program Files (x86)\\Nmap\\nmap.exe" # Update this if needed

def scan\_ports():

target = entry\_target.get().strip()

if not target:

messagebox.showerror("Error", "Please enter a target IP or hostname")

return

# Check if Nmap exists

if not os.path.exists(NMAP\_PATH):

messagebox.showerror("Error", f"Nmap not found at {NMAP\_PATH}. Please check the installation path.")

return

scanner = nmap.PortScanner(nmap\_search\_path=(NMAP\_PATH,)) # Explicitly set Nmap path

try:

scanner.scan(hosts=target, arguments='-p 1-1024 -T4')

result\_text.delete("1.0", tk.END)

if not scanner.all\_hosts():

result\_text.insert(tk.END, "No hosts found or scan failed. Check target IP and permissions.\n")

return

for host in scanner.all\_hosts():

result\_text.insert(tk.END, f"Scanning {host} ({scanner[host].hostname()})\n")

for proto in scanner[host].all\_protocols():

result\_text.insert(tk.END, f"Protocol: {proto}\n")

ports = scanner[host][proto].keys()

if not ports:

result\_text.insert(tk.END, "No open ports found.\n")

for port in sorted(ports):

state = scanner[host][proto][port]['state']

result\_text.insert(tk.END, f"Port {port}: {state}\n")

except Exception as e:

messagebox.showerror("Error", f"Nmap scan failed: {str(e)}")

# GUI Setup

root = tk.Tk()

root.title("Nmap TCP Scanner")

root.geometry("500x400")

tk.Label(root, text="Enter Target IP/Hostname:").pack()

entry\_target = tk.Entry(root, width=50)

entry\_target.pack()

scan\_button = tk.Button(root, text="Scan", command=scan\_ports)

scan\_button.pack()

result\_text = tk.Text(root, height=15, width=60)

result\_text.pack()

root.mainloop()

**Results and Observations:** The application successfully scans the specified target for open TCP ports in the range 1-1024. The scan results display the protocol, open ports, and their status, allowing users to analyze network security.

**Conclusion:** This project demonstrates a practical application of network security by integrating Nmap scanning within a Python GUI. The tool simplifies port scanning and helps users assess vulnerabilities in networked systems.

**Future Enhancements:**

* Expand port range scanning options.
* Implement additional scanning techniques such as UDP scanning.
* Improve UI design for better user experience.

**References:**

1. Fyodor, "Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning."
2. Nmap Documentation: <https://nmap.org/docs.html>