```
Project Title: Port Scanner
Objective: Scan open ports on a system to detect potential vulnerabilities.
Tech Stack: Python, Nmap API, Flask.
Features:
- Detect open and closed ports.
- Identify running services and protocols.
- Generate scan reports.
- Bonus: Add multi-threading for faster scans.

--------------------
port_scanner_assignment.py
--------------------

from flask import Flask, request, jsonify
from threading import Thread
import nmap

app = Flask(__name__)
scan_results = {}

def scan_ports(target, port_range='1-1024'):
    scanner = nmap.PortScanner()
    try:
        scanner.scan(hosts=target, ports=port_range, arguments='-sS -T4')
        result = {}
        for host in scanner.all_hosts():
            result[host] = []
            for proto in scanner[host].all_protocols():
                ports = scanner[host][proto].keys()
                for port in ports:
                    service = scanner[host][proto][port]['name']
                    state = scanner[host][proto][port]['state']
                    result[host].append({
                        'port': port,
                        'state': state,
                        'service': service,
                        'protocol': proto
                    })
        return result
    except Exception as e:
        return {"error": str(e)}

@app.route('/')
def index():
    return '''
    <html>
    <head><title>Port Scanner</title></head>
    <body>
        <h2>Port Scanner Assignment</h2>
        <form method="POST" action="/scan">
            Target IP / Domain: <input type="text" name="target" required><br><br>
                    Port  Range  (e.g.,  1-1024):  <input  type="text"  name="port_range"
value="1-1024"><br><br>
            <input type="submit" value="Start Scan">
```

```
        </form>
    </body>
    </html>
    '''

@app.route('/scan', methods=['POST'])
def scan():
    target = request.form['target']
    port_range = request.form.get('port_range', '1-1024')
    def run_scan():
        global scan_results
        scan_results[target] = scan_ports(target, port_range)
    thread = Thread(target=run_scan)
    thread.start()
      return f"Scan for {target} started. Please check /result/{target} after a few seconds."

@app.route('/result/<target>')
def result(target):
       return jsonify(scan_results.get(target, {"message": "Scan is running or not found."}))

if __name__ == '__main__':
    app.run(debug=True)
```