Cyber3DMap-Streamlit

A penetration testing tool for visualizing network attack surfaces in a GNS3-like 3D map. Built with Streamlit, NetworkX, Plotly, and PyTorch Geometric, it processes Nmap XML scans to map vulnerabilities, predict attack paths, and simulate real-time threats.
Features

Interactive 3D Map: Visualize up to 10 nodes with IPs, services, and CVEs.
CVE Visualization: Nodes colored by CVSS severity:
Red: High risk (>7, e.g., SMBGhost CVE-2020-0796).
Yellow: Medium risk (>4–7, e.g., MySQL CVE-2019-3738).
Green: Low risk (≤4, e.g., vsftpd 3.0.3).
Purple: Active threat (cycles every 4s).

Attack Path Prediction: Graph Neural Network (GNN) prioritizes exploitable nodes.
Real-Time Threat Simulation: Models attacker movement.
Modular Configuration: Customize via config.yaml.

Warning
⚠ Limit to 10 Nodes: Processing >10 nodes may cause delays due to NVD API rate limits and rendering complexity. Segment large scans (e.g., 192.168.1.1-10).
Project Structure

src/
app.py: Streamlit app for scan uploads and visualization.
configs/config.yaml: Settings for GNN, colors, and API.
core/
parser.py: Parses Nmap XML into nodes/edges.
cve_fetcher.py: Fetches CVEs with caching.
graph_manager.py: Manages NetworkX graph.
gnn_model.py: GNN for attack path prediction.
visualizer.py: Renders 3D map.

requirements.txt: Dependencies.

data/graph.json: Sample graph.
examples/big_scan.xml: Sample 10-node Nmap scan.

Script Functions

app.py: Runs the UI, coordinates scan processing, visualization, and predictions.
config.yaml: Configures GNN layers, node colors, and API rate limits.
parser.py: Extracts IPs, services, and ports from Nmap XML.
cve_fetcher.py: Queries NVD API, caches CVEs, handles retries.
graph_manager.py: Stores/retrieves graph with CVE data.
gnn_model.py: Predicts attack paths using GNN.
visualizer.py: Generates GNS3-like 3D map with Plotly.

Setup

Prerequisites: Python 3.10+, internet for NVD API.
Clone Repository:git clone https://github.com/SunnyThakur25/Cyber3DMap-Streamlit.git
cd Cyber3DMap-Streamlit

Install Dependencies:cd src
pip install -r requirements.txt

Run:streamlit run app.py

Usage

Scan Network (≤10 hosts):nmap -sV -oX examples/scan.xml 192.168.1.1-10

Upload Scan:
Open http://localhost:8501.
Upload examples/scan.xml.

Analyze:
Red Nodes: Exploit high-severity CVEs (e.g., msfconsole for SMBGhost).
Yellow Nodes: Test for escalation (e.g., hydra for MySQL).
Green Nodes: Verify configs (e.g., ftp for vsftpd).
Purple Node: Simulate lateral movement.
Hover for details (IP, ports, CVEs).

Predict Paths:
Click “Predict Attack Paths” for GNN scores.

Report:
Screenshot map, copy JSON from “Network Details”.

Pentesting Workflow

Scan: Run Nmap for ≤10 hosts.
Visualize: Upload XML to map vulnerabilities.
Exploit: Target red nodes (e.g., CVE-2020-0796).
Simulate: Use purple nodes for pivoting.
Predict: Prioritize paths with GNN.
Report: Document with map and JSON.

Troubleshooting

No Map: Check UI/terminal errors. Clear cache:del %USERPROFILE%.streamlit\cache

Slow CVEs: Set rate_limit: 2 in config.yaml.
Invalid XML: Validate:python -c "import xml.etree.ElementTree as ET; ET.parse('examples/scan.xml')"

GNN Issues: Confirm torch==2.4.1, torch-geometric==2.5.3.

Security

Obtain client permission for scans.
Delete scan files post-use:del examples\scan.xml

Encrypt sensitive data.

Limitations

Optimized for ≤10 nodes.
NVD API delays possible.
GNN accuracy depends on scan quality.

License
MIT License. See LICENSE.
Contributing
Fork, create a branch, and submit a pull request. Issues and feature requests welcome!
Contact

GitHub: SunnyThakur25
Email: sunny48445@gmail.com

⭐ Star this repo if you find it useful!