

Florida International University

Summer 2025 Senior Design Project



Frozen Network Reporting

Team Members: Kevin Ocana, Anthony Torres, Marek Ferfecky, Elizabeth Luzardo

Instructor/Faculty: *Masoud Sadjadi* Florida International University

What is Frozen Network Reporting?

FNR is a network vulnerability scanner designed to automatically detect security weaknesses in devices, open ports, and services. Using a combination of ICMP ping sweeps, TCP/UDP port scanning, and service fingerprinting, FNR identifies potential entry points for cyberattacks.



Why we created a Vulnerability Scanner

Networks are constantly targeted by malicious actors aiming to exploit vulnerabilities and compromise security. The consequences of these threats include, but are not limited to:

- Unauthorized access to sensitive data
- Disruption of critical services
- Deployment of malware or ransomware
- Financial loss and reputational damage

These threats often originate from various sources, such as:

- Unpatched software or outdated systems
- Misconfigured devices
- Rogue or unauthorized devices
- Malicious external scans and attacks



Networking Vulnerabilities: why they happen and how to respond

Organizations often fall prey to network threats and vulnerabilities due to a lack of visibility or awareness of common attack vectors that include but are not limited to:

- Misconfigured Firewalls, Services, and Ports
- Outdated Software
- Insecure Wireless Networks
- Malware
- Social engineering by phishing
- Malicious links
- Unsecured devices

What to do if a network is compromised:

- Conduct a vulnerability scan
- Isolate affected systems and investigate access logs
- Patch known vulnerabilities with updates
- Follow incident response protocols



Tools/Libraries

Nmap

- A powerful open-source tool Nmap scans the ports of network devices and probes website domains for known vulnerabilities



OpenVAS

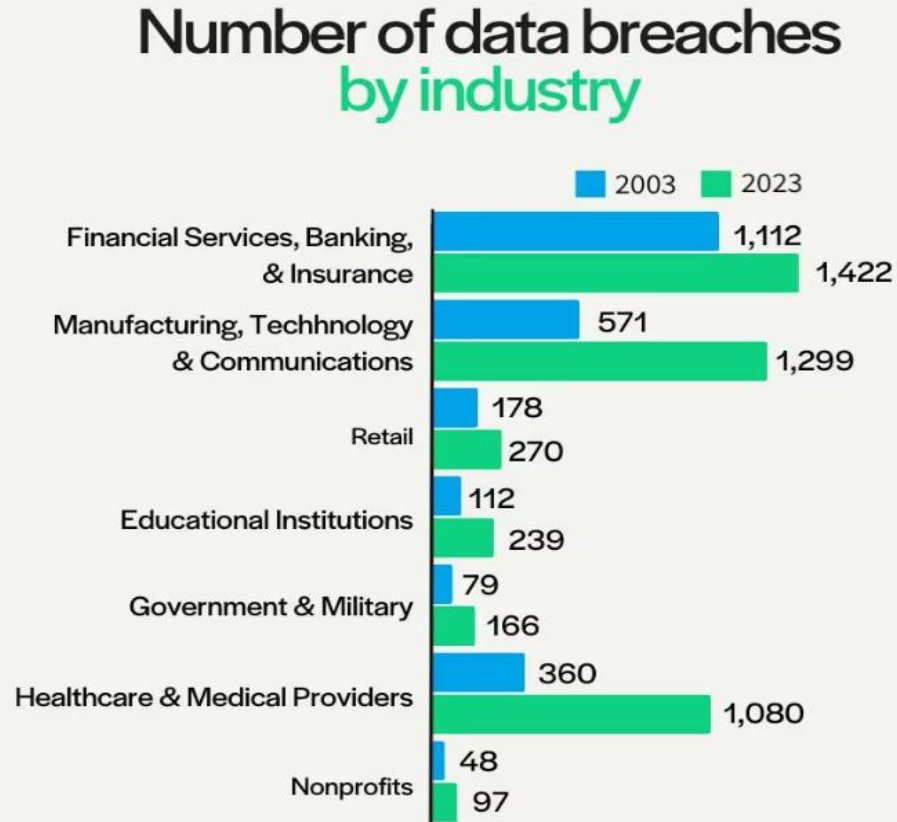
- A full-featured vulnerability scanner that identifies security issues in systems and applications.

Kali Linux

- Users can use the command “apt-get install Nmap” to install it.



Data Breaches



Source: Privacy Rights Clearinghouse, 2023

secureframe

- By 2025, the global cost of cybercrime is projected to reach \$10.5 trillion, growing at a rate of 15 percent annually
- 45% of Americans have had their personal information compromised by a data breach in the last five years
- 82% of data breaches involve data stored in the cloud. 39% of breaches span multiple environments and incur a higher-than-average data breach cost of \$4.75 million

- DDoS attacks spiked in 2024 with approximately 8 million DDoS attacks taking place within the first half of the year. According to NetScout
- the average annual cost of cybercrime worldwide is expected to soar from \$8.4 trillion in 2022 to more than 23 trillion in 2027 (Neuberger, 2023).
- Worm malware was the most commonly prevented type of malware in 2022 with ~210 million incidents occurring. Virus malware followed in second, with ~200 million detections, followed by ransomware with over 106 million.
- 495 million ransomware attempts were detected globally in 2022.

Examples of IoCs

Misconfiguration:

Having an inbound rule set allowing 0.0.0.0/0 through Port 80

```
C:\>kubectl apply -f sample-network-policy.yaml
networkpolicy.networking.k8s.io/sample-network-policy configured

C:\>kubectl describe networkpolicy sample-network-policy
Name:         sample-network-policy
Namespace:    default
Created on:   2022-10-30 18:16:43 -0700 PDT
Labels:       <none>
Annotations:  <none>
Spec:
  PodSelector:  app=webapp
  Allowing ingress traffic:
    To Port: 443/TCP
    To Port: 80/TCP
    From:
      PodSelector: app=webapp
  -----
  To Port: <any> (traffic allowed to all ports)
  From:
    PodSelector: <none>
  -----
  To Port: 443/TCP
  To Port: 80/TCP
  From:
    IPBlock:
      CIDR: 172.16.0.0/16
      Except:
  -----
  To Port: 443/TCP
  To Port: 80/TCP
  From:
    IPBlock:
      CIDR: 0.0.0.0/0
      Except:
```

Outdated Software:

Having an Outdated Windows 2012 server installed that reached end of life support.



Insecure Networking:

Weak encryption or passwords set for a Wireless Network



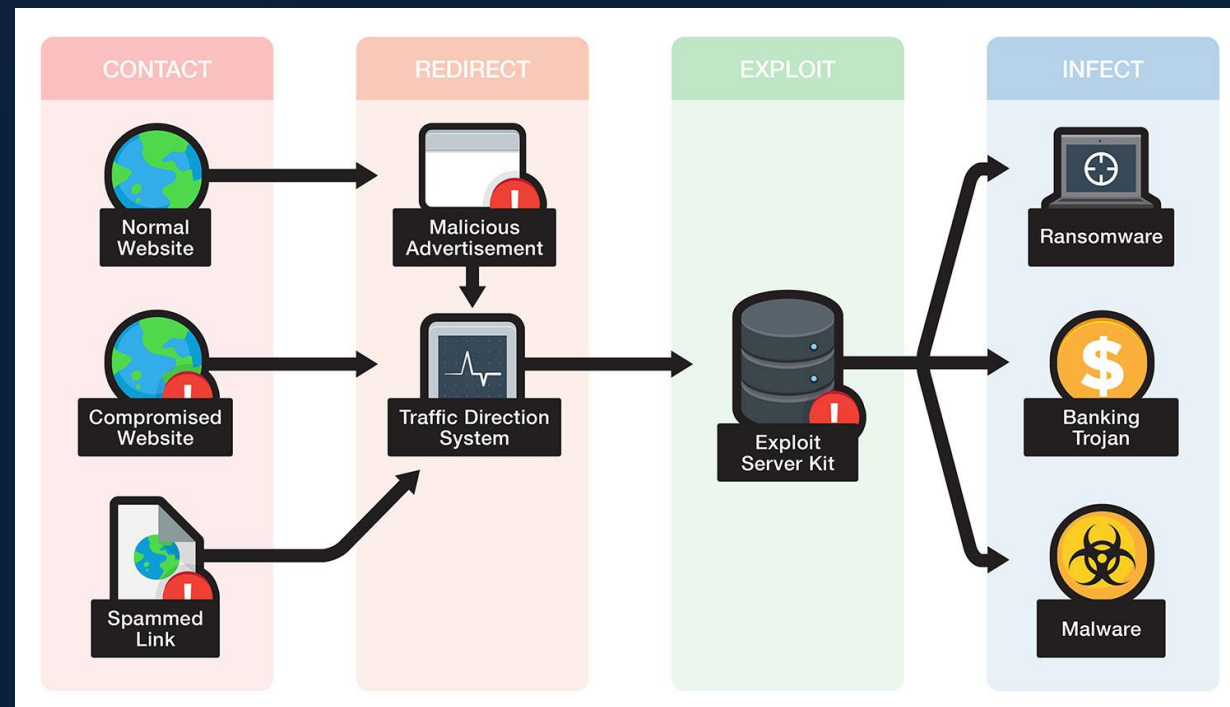
Malware:

Detecting a worm replicating throughout a network.



How Network Exploits work

- Network exploits occur when attackers identify and take advantage of vulnerabilities such as outdated software, misconfigured devices, or open ports. Once access is established, they are able to move through the system and steal data or disrupt operations. A network vulnerability scanner can detect this access early and allow the organizations to take preventative actions before an attack can succeed.



How the scanner works



- Frontend:
 - Would have an interface with HTML, CSS, or JavaScript
 - Would have different visuals for criticality for key findings/vulnerabilities for what was scanned.
 - Functions to allow dismissal, setup notifications,
 - Runs on user's browser
 - API requests
- Backend:
 - Built in languages like Python or Java
 - Frameworks like Flask or Django
 - Uses tools like Nmap, Scapy or system socket calls
 - Runs on Server or cloud
 - API response
- Scanning:
 - Automated daily port scanning
- Data visualization:
 - Chart.js

Drawbacks, limitations, and concerns

Privacy:

- The scanner will only analyze public network information to ensure no sensitive information is collected.

Accuracy:

- False positive and false negatives may occur which could lead to missed vulnerabilities or alerts.
- New threats and attacks may not be detected until the scanner is updated to address those vulnerabilities
- Changes in network protocols or infrastructure could impact the scanners' ability to access or analyze data.

Verification & Testing

Network Vulnerability Scanner

HOME PORTS INFORMATION SERVICE ABOUT CONTACT US

Target IP or Hostname:

Select Ports to Scan:

21 22 23 25
53 80 110 139
143 443 445 3389
3306 8080 5900 1723
8000 8443

Scan

Selected Target and Ports

Target: scanme.nmap.org
Ports: 80, 143, 443, 3389, 3306, 1723

Scan Results

Port	Protocol	State	Service	Version	Vulnerabilities (CVEs)
80	tcp	open	http	2.4.7	No known CVEs found.
143	tcp	closed	imap		No known CVEs found.
443	tcp	filtered	https		No known CVEs found.
1723	tcp	closed	pptp		No known CVEs found.
3306	tcp	closed	mysql		No known CVEs found.
3389	tcp	closed	ms-wbt-server		No known CVEs found.

FROZEN NETWORK REPORTING

Network Vulnerability Scanner

HOME PORTS INFORMATION SERVICE ABOUT CONTACT US

Top 50 Most Common Ports Information:

Port Number	Transport Protocol	Description	Information
20	TCP	FTP Data	Used for transferring data in FTP sessions.
21	TCP	FTP Control	Controls FTP commands and authentication.
22	TCP	SSH	Secure shell for remote logins and command execution.
23	TCP	Telnet	Legacy remote login service, not encrypted.
25	TCP	SMTP	Sends email between servers.
53	UDP/TCP	DNS	Resolves domain names to IP addresses.
67	UDP	DHCP Server	Assigns IP addresses to devices.
68	UDP	DHCP Client	Receives IP addresses from DHCP server.
69	UDP	TFTP	Simplified file transfer protocol.
80	TCP	HTTP	Standard web traffic.
110	TCP	POP3	Retrieves emails from mail servers.
119	TCP	NNTP	Used for reading and posting Usenet articles.
123	UDP	NTP	Time synchronization across devices.
135	TCP	RPC	Remote Procedure Call for Windows services.
137	UDP	NetBIOS Name Service	Name resolution in Windows networks.
138	UDP	NetBIOS Datagram Service	Browser and messaging in NetBIOS.
139	TCP	NetBIOS Session Service	File and printer sharing in Windows.
143	TCP	IMAP	Access email on a mail server.
161	UDP	SNMP	Network management monitoring.
162	UDP	SNMP Trap	SNMP alert messages.
179	TCP	BGP	Routing information between autonomous systems.

Code example of



Example of Python code to scan IP's for open ports:

- Import sockets provides a way to interact with endpoints of communication in a network.
- Import nmap allows the user to interact with the Nmap port scanner, enabling the automation of scanning tasks and the parsing of Nmap scan results

```
import socket
import nmap

def port_scan(target_ip, ports):
    open_ports = []
    for port in ports:
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(1) # Timeout to avoid hanging
            result = sock.connect_ex((target_ip, port))
            if result == 0:
                open_ports.append(port)
            sock.close()
        except Exception as e:
            print(f"Error scanning port {port}: {e}")
    return open_ports
```

Test Suite Steps

UI/UX Testing

1. Dataset Preparation
 - Labeled dataset of network devices (safe/vulnerable)
 - Include edge cases (firewalls, IoT)
2. Test to Perform
 - Admin feedback on report clarity.

Functional Testing

1. Core Functionality
 - Test port scanning (TCP/UDP)
 - Service/OS detection accuracy
2. Vulnerability Matching
 - Verify CVE database lookups
 - False positive/negative checks
 - Validate risk scoring logic

Performance and Security Testing

1. Stress Testing
 - High volume target scans (100+ IPs)
 - Resource usage (CPU/RAM during scans)
2. Stealth Testing
 - Detectability by IDS/IPS

Test Cases

Test Case	Test description	Input	Expected Output	Result
Port Scan Accuracy	Verify correct detection of open TCP/UDP ports	Scan 192.168.1.1-100 (preconfigured)	100% match with known open or closed ports	Pass/Fail
Vulnerability Matching	Validate CVE database lookups	Scan a device with known vulnerabilities according to CVE	Correctly flags the vulnerability with correct risk level	Pass/Fail
Stress Test	Handle high volume scans	Simultaneously scan 500+ IPs	No crashes, CPU usage <80%	Pass/Fail

Brainstorming Ideas

Should we build a completely custom scanning algorithm to identify common network vulnerabilities?
Should we publish the app to be more accessible for others to use?
Should there be features such as scheduled scans, log exports, and real-time alerts to improve the effectiveness of the application?

How The App Works

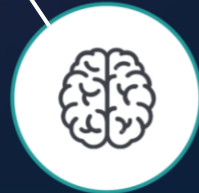
Python based network scanner that identifies security weakness in target systems through a multistage scanning process. Uses ICMP ping sweeps and ARP scanning to map connected hosts. Then performs comprehensive port scanning using both TCP and UDP protocols to detect open ports and services running on each device. For deeper analysis, the scanner cross references it's findings with the CVE database to identify known vulnerabilities. The GUI is constructed using Streamlit a simple and effective web-based GUI to make scanning as simple as possible. The results are then converted into PDF format for easy extract.

Post Prototype Review

Address all vulnerabilities
Decide whether the prototype is ready for production or if it needs more testing. If the prototype needs more testing, we will repeat the previous step with corrections until ready.



Frozen Network
Scanning



Why We Created The App

With the growing number of network attacks and system breaches, organizations struggle to keep up with evolving threats. Vulnerabilities such as outdated software and open ports are exploited everyday without detection. We developed this vulnerability scanner to identify and mitigate these risks, helping organizations secure their infrastructure and preventing damaging intrusions before they occur.

Narrowing Down Ideas

We decided to build our network vulnerability scanner using python, due to its extensive libraries and our teams existing proficiency with the language. This will allow us to efficiently implement scanning and automate testing procedures. By creating our own detection software, we can maintain how vulnerabilities are detected and focus on common issues.

Testing The Prototype

Functional tests, with several example IPs will be undergone to ensure accurate port scanning throughout vulnerability detection. UI and UX will be ensured through Admin feedback of usability and clarity of report generated, as well as data set refinement so that CVE matches vulnerabilities. Finally, performance will be tested through a stress test on the capabilities of concurrent scans at one time while still maintaining crash free and low CPU usage.

References

- Kime, C. (2024b, September 13). *How to use NMAP for vulnerability scanning: complete tutorial*. eSecurity Planet. <https://www.esecurityplanet.com/networks/nmap-vulnerability-scanning-made-easy/>
- Bonnie, E. (2025, January 3). 110+ of the latest data breach statistics [Updated 2025]. *Secureframe*. <https://secureframe.com/blog/data-breach-statistics>
- *Technical difficulties*. (n.d.). <https://2021-2025.state.gov/digital-press-briefing-with-anne-neuberger-deputy-national-security-advisor-for-cyber-and-emerging-technologies/>
- Gmcdouga. (2024). A Closer Look at Q3 2024: 75% Surge in Cyber Attacks Worldwide. Retrieved from <https://blog.checkpoint.com/research/a-closer-look-at-q3-2024-75-surge-in-cyber-attacks-worldwide/>
- Kerner, S. M. (2025). 35 cybersecurity statistics to lose sleep over in 2025. Retrieved from <https://www.techtarget.com/whatis/34-Cybersecurity-Statistics-to-Lose-Sleep-Over-in-2020>
- Dilmegani, C. (2025). *100+ Network Security Statistics in 2025*. AIMultiple. Retrieved from <https://research.aimultiple.com/network-security-statistics/>