Capstone Project
Cybersecurity Architecture & Threat Mitigation

Chukwuebuka Tobiloba Nwaizugbe
Email: nwaizugbechukwuebuka@gmail.com
LinkedIn: linkedin.com/in/chukwuebuka-tobiloba-nwaizugbe

May 2025

Abstract

This capstone project demonstrates professional expertise in designing cybersecurity architectures and threat mitigation aligned with ISO 27001 and the NIST Cybersecurity Framework. It includes vulnerability assessments, threat detection, and automated malware analysis, showcasing my ability to build secure enterprise environments and utilize advanced security tools. This experience positions me for demanding roles in cybersecurity operations, threat intelligence, and security architecture.

1 Executive Summary

This project developed an ISO 27001 and NIST-aligned cybersecurity architecture for a simulated enterprise environment. Utilizing tools such as Nessus, Wireshark, Snort, and VirusTotal, I identified and remediated over 30 critical vulnerabilities. A Python-based malware detector automates threat detection through integration with threat intelligence APIs, highlighting innovation and programming proficiency within cybersecurity.

2 Project Objectives

- Develop a robust cybersecurity framework based on ISO 27001 and NIST CSF.
- Conduct comprehensive vulnerability and network traffic analyses to minimize risk.
- Create an automated malware detection system in Python leveraging Snort signatures and VirusTotal.
- Build practical expertise in cybersecurity tools to prepare for roles in threat mitigation and architecture.

3 Methodology

3.1 Cybersecurity Architecture

Designed a multi-layered security model featuring network segmentation, strict access controls, endpoint protection, and incident response protocols. The framework adheres to ISO 27001 and NIST CSF requirements, ensuring compliance and scalability.

3.2 Vulnerability and Network Traffic Analysis

Performed extensive vulnerability scans using Nessus, identifying over 30 security gaps including misconfigurations and outdated services. Employed Wireshark to analyze network traffic, detect anomalies, and address insecure protocols.

3.3 Malware Detection System Development

Developed a Python-based malware detector that calculates SHA256 hashes and queries the VirusTotal API for threat assessment. Implemented a simulated Snort rule engine to identify known malware patterns, combining signature- and heuristic-based detection methods.

4 Results and Impact

- Delivered an ISO and NIST-compliant cybersecurity framework securing critical assets.
- Remediated 30+ vulnerabilities, significantly reducing attack surface.
- Built a fully operational malware detection tool demonstrating advanced API integration.
- Enhanced practical skills in vulnerability management, network forensics, and malware analysis.

5 Challenges and Professional Growth

- Navigated complex international security standards, strengthening attention to detail and critical thinking.
- Implemented robust error handling and API rate limiting to improve tool reliability.
- Gained real-world insights into intrusion detection via Snort simulation.
- Deepened understanding of layered defense and proactive threat mitigation strategies.

6 Conclusion and Career Relevance

This project demonstrates my advanced technical capabilities and professional maturity in cybersecurity. By integrating architecture design with automated threat detection, I am prepared to contribute effectively to security operations, threat intelligence, and risk management. Future work will explore machine learning and real-time network analysis to address evolving cybersecurity challenges.

7 References

- ISO/IEC 27001:2013 Information Security Management
- NIST Cybersecurity Framework (CSF)
- Nessus Vulnerability Scanner Documentation
- Wireshark Network Protocol Analyzer
- Snort Intrusion Detection System
- VirusTotal API Documentation
- Python Official Documentation

Appendices

- Code excerpts and usage instructions
- Network architecture diagrams
- Detailed vulnerability scan reports