A

Project Proposal on

Automated Security Posture Analysis

submitted by

Mr. Siddhesh Santosh Shinde

In partial fulfilment of the award of the degree

of

Master of Science

in

Computer Science

under the guidance of

Mrs. Priya Katdare

Department of Computer Science

Dr. Datar Science, Dr. Behere Arts, Shri. Pilukaka Joshi Commerce College (Autonomous), Chiplun,

(D. B. J. College, Chiplun)

Academic year 2025-2026

Navkonkan Education Society's

Dr. Datar Science, Dr. Behere Arts, Shri. Pilukaka Joshi Commerce College (Autonomous), Chiplun, (D. B. J. College, Chiplun)

==========

Department Of Computer Science

CERTIFICATE

This is to certify that **Mr. Siddhesh Santosh Shinde** of **MSc. PART-II Computer Science** class has satisfactorily completed the Project **"Automated Security Posture Analysis"**, to be submitted in partial fulfilment of the award of **Master of Science** in **Computer Science** during the academic year **2025 – 2026**.

Project Guide **Head / In-Charge,**

(Department of Computer Science)

Date of Submission:

College Seal **Signature of Examiner**

DECLARATION

I, **Siddhesh Santosh Shinde**, a student of **D.B.J College**, pursuing the **Master of Science in Computer Science** program, hereby declare project titled **"Automated Security Posture Analysis"** is an original work carried out by me.

I further declare that this work has not been previously submitted for any other purpose, examination, or assessment. I also assure you that all the sources and references used in this project have been duly acknowledged.

I assure you that this research report is a result of my efforts and represents my understanding of the subject matter.

Signature of the Student:

Siddhesh Santosh Shinde

Place: Chiplun

Date:

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide, **Mrs Priya Katdare**, for their guidance and support during the preparation of this project proposal.

I also thank the **Department of Computer Science** and the faculty members of **D.B.J. College (Autonomous), Chiplun,** for providing the required resources and a supportive environment.

Finally, I am thankful to everyone who directly or indirectly contributed to the successful completion of this proposal.

Siddhesh Shinde (M.Sc Part-II)

TABLE OF CONTENTS

Sr. No	Contents	Page No.
1.	Title	1
2.	Objective	2
3.	Introduction	3
4.	Literature Survey	4
5.	Methodology	7
6.	Significance and Scope of Work	10
7.	Conclusion	12
8.	References	14

Automated Security Posture Analysis

OBJECTIVE

The primary objectives of this project are:

1. **Evaluate System Defenses:**
   Assess the effectiveness of critical security controls such as Antivirus (AV), Endpoint Protection, Firewall, and Email Security by analyzing system configurations and network posture.

2. **Automated Risk Assessment:**
   Provide a defense score based on the current security posture of a system, highlighting strengths and weaknesses in protection against potential cyber threats.

3. **Recommendations for Improvement:**
   Generate actionable, data-driven recommendations to strengthen defenses where gaps are identified, ensuring proactive mitigation of risks.

4. **Simplified User Interaction:**
   Enable users to input an IP address and receive a comprehensive security posture analysis without requiring deep technical expertise.

5. **Support Cybersecurity Readiness:**
   Assist organizations and individuals in maintaining compliance with security benchmarks (e.g., CIS, NIST) and improving resilience against hacker attempts.

INTRODUCTION

Cybersecurity has become one of the most critical concerns in today's digital era. Organizations and individuals face constant threats from hackers who exploit weaknesses in system defenses such as antivirus software, endpoint protection, firewalls, and email security. While traditional security audits are often manual, time-consuming, and require specialized expertise, attackers continue to evolve their techniques, making automated and intelligent defense evaluation essential.

The **Automated Security Posture Analyzer (ASPA)** project addresses this challenge by leveraging **Python, Machine Learning, and Artificial Intelligence** to evaluate the effectiveness of system defenses. By simply providing an IP address, users can quickly assess whether their security controls are properly configured and active. The tool generates a **defense score**, highlights vulnerabilities, and provides actionable recommendations to strengthen the overall security posture.

Key challenges in cybersecurity defense evaluation include:

- **Heterogeneous environments:** Systems may run different operating systems, configurations, and security tools.
- **Dynamic threats:** Attack techniques evolve rapidly, requiring adaptive and intelligent analysis.
- **Complex dependencies:** Security often relies on multiple layers (AV, endpoint, email, firewall), which must work together seamlessly.
- **Accessibility:** Non-technical users need simplified tools to understand their exposure without deep expertise.

By integrating automated scanning, ML-based risk classification, and recommendation generation, ASPA aims to make **cybersecurity posture assessment faster, more accurate, and more accessible**. This project not only supports proactive defense but also helps organizations align with security benchmarks such as **CIS Controls** and **NIST Cybersecurity Framework**.

LITERATURE SURVEY

Machine learning and AI in cybersecurity posture assessment

- **Detection beyond signatures:**
  ML models (e.g., tree ensembles, anomaly detection) augment traditional signature-based approaches by learning patterns of malicious behavior in network flows, process activity, and file attributes. This improves detection of zero-day and polymorphic threats without relying solely on known indicators.
- **Risk scoring and prioritization:**
  Supervised models and heuristic scoring systems combine signals (open ports, outdated services, weak configurations, email records, endpoint telemetry) into composite risk scores, enabling operations teams to triage and address the highest-impact weaknesses first.
- **Behavioral analytics:**
  Unsupervised methods (e.g., clustering, isolation forests) identify shifts in normal baselines for endpoints and users, flagging anomalies such as unusual process trees, login patterns, or lateral movement precursors.
- **Explainability and trust:**
  Techniques like SHAP and feature importance are used to interpret model outputs, helping analysts understand why a system is rated as high risk and mapping findings to concrete remediation actions.

Endpoint protection and antivirus effectiveness

- **Telemetry-driven assessment:**
  Studies highlight the value of endpoint telemetry (process creation, module loads, registry changes, kernel callbacks) for detecting malicious activity that AV may miss. Combining AV status (installed, updated, real-time protection on) with telemetry-based anomaly detection yields stronger posture signals.
- **Defense-in-depth on endpoints:**
  Hardened configurations (application control, exploit protection, ASR rules, EDR sensors) significantly reduce compromise risk. Literature emphasizes the importance of patch compliance, least privilege, and controlled software inventory alongside AV.
- **Common failure modes:**
  Endpoints often fail due to disabled AV or outdated signatures, misconfigured firewalls, weak PowerShell/Office macro policies, and stale EDR agents. Automated checks for these states are consistently recommended.

Network exposure, configuration hygiene, and vulnerability scanning

- **Service discovery and exposure:**
  Network scanning (e.g., Nmap-like capabilities) remains foundational for discovering exposed services, obsolete protocols, and misconfigurations. Research shows that open management ports, legacy SSL/TLS, and default credentials are frequent root causes of breaches.
- **Configuration baselines:**
  Mapping host and network settings to recognized baselines (CIS Benchmarks, NIST CSF) provides a structured posture lens. Automated compliance checks against these controls improve consistency and repeatability.
- **TLS and crypto hygiene:**
  Analyses consistently link weak cipher suites, outdated TLS versions, and missing HSTS to elevated risk, recommending automated tests and clear remediation guidance.

Email security authentication and anti-phishing defenses

- **SPF, DKIM, DMARC impacts:**
  Literature underscores that properly configured SPF/DKIM/DMARC materially reduces spoofing and brand abuse. Automated DNS checks for these records, plus alignment and enforcement modes, are central to posture scoring.
- **Defense layers:**
  Beyond authentication records, effective anti-phishing requires attachment and URL inspection, sandboxing, user awareness training, and impersonation protection. ML-based classifiers assist by flagging phishing indicators (language cues, sender anomalies, domain age).
- **Common gaps:**
  Missing DMARC enforcement, misaligned DKIM, permissive SPF, and lack of impersonation controls are frequent issues; surveys recommend routine audits and dashboards that track domain posture.

Posture scoring frameworks and operationalization

- **Composite scoring models:**
  Research favors weighted scoring frameworks that combine endpoint, network, and email signals into a single, interpretable score. Calibration against incident histories or red-team findings improves relevance.
- **Actionable remediation mapping:**
  High-quality posture systems pair findings with prioritized playbooks—clear, low-friction steps that can be tracked to closure. This increases adoption and real-world risk reduction.
- **Continuous assessment and drift detection:**
  Literature recommends periodic or continuous assessment, alerting on drift

from secure baselines (e.g., a firewall disabled after an update), and integrating with ticketing/SIEM for workflow automation

METHODOLOGY

This project aims to design and implement an **Automated Security Posture Analyzer (ASPA)** that evaluates system defenses against potential hacker threats. The methodology integrates Python scripting, machine learning, and AI-driven recommendations to provide a comprehensive security assessment.

Project Overview

The tool will analyze system defenses such as **Antivirus (AV), Endpoint Protection, Firewall, and Email Security** by scanning configurations and network posture. Based on the findings, ASPA will generate a **defense score** and provide **recommendations** for strengthening weak or missing protections.

◇ Steps in Methodology

1. **Environment Setup**
    - Install Python and required libraries (nmap, psutil, dnspython, scikit-learn, streamlit).
    - Configure a virtual environment for modular development.
2. **Input Phase**
    - User provides an IP address through a simple interface (CLI or Streamlit dashboard).
    - The system validates the IP format before initiating scans.
3. **Data Collection & Scanning**
    - Antivirus Check: Detect if AV is installed, updated, and active.
    - Endpoint Protection: Verify firewall status, patch compliance, IDS/IPS presence.
    - Email Security: Query DNS records for SPF, DKIM, and DMARC validation.
    - Network Exposure: Perform port scans, detect open services, and check TLS/SSL configurations.
4. **Data Preprocessing**
    - Normalize collected data into structured features (e.g., AV status = Active/Inactive, Firewall = Enabled/Disabled).
    - Encode categorical values for ML model input.
5. **Analysis & Prediction**
    - Apply **rule-based checks** for baseline compliance (CIS/NIST standards).
    - Use **ML classification models** (e.g., Random Forest, Logistic Regression) to predict overall risk level (Secure / Vulnerable).
    - Generate a **defense score (0–100)** based on weighted factors.

6. **Recommendation Engine**
    - Map detected weaknesses to actionable remediation steps.
    - Example:
        - If AV not detected → "Install and enable a trusted antivirus solution."
        - If firewall disabled → "Enable host-based firewall and restrict unnecessary ports."
        - If SPF/DKIM missing → "Configure SPF/DKIM/DMARC to prevent email spoofing."
7. **Reporting & Visualization**
    - Display results in a **Streamlit dashboard** with:
      Security score
      Active vs inactive defenses
      Recommendations list
    - Export findings into a structured report for documentation.

◇ Aim of the Project

- Provide **automated, real-time assessment** of system defenses.
- Simplify cybersecurity posture evaluation for non-experts.
- Support organizations in aligning with **CIS Controls** and **NIST Cybersecurity Framework**.
- Deliver actionable insights to improve resilience against hacker attempts.

◇ Libraries and Framework Used:

The implementation of the Automated Security Posture Analyzer leverages a combination of Python libraries and frameworks to perform scanning, analysis, machine learning, and visualization.

◆ Core Libraries

- **Python Standard Library**
    - socket → For basic IP and port connectivity checks.
    - os / subprocess → For system-level queries and command execution.
- **psutil**
    - Used to monitor system processes and services.
    - Helps detect whether Antivirus and Firewall services are running.
- **nmap (python-nmap)**
    - Provides network scanning capabilities.
    - Detects open ports, running services, and potential vulnerabilities.
- **dnspython**

- Used for DNS queries to validate email security records (SPF, DKIM, DMARC).

◇ Data Handling & Preprocessing

- **Pandas**
  - For structured data manipulation and tabular representation of scan results.
- **NumPy**
  - For numerical computations and feature normalization.

◇ Machine Learning & AI

- **scikit-learn**
  - Provides ML algorithms for classification (Logistic Regression, Random Forest).
  - Includes utilities for train/test split, feature scaling, and evaluation metrics.
- **SHAP / Feature Importance (optional)**
  - For model interpretability and explaining why a system is rated secure/insecure.

◇ Visualization & Reporting

- **Matplotlib / Seaborn**
  - For plotting defense scores, confusion matrices, and feature importance charts.
- **Streamlit**
  - Framework for building an interactive web-based dashboard.
  - Allows users to input IP addresses and view results in real time.

◇ Development Environment

- **Visual Studio Code**
  - Used as the primary IDE for coding, debugging, and project organization.
- **Virtual Environment (venv/conda)**
  - Ensures isolated dependency management for reproducibility.

SIGNIFICANCE AND SCOPE OF WORK

The scope of the **Automated Security Posture Analyzer (ASPA)** extends beyond simple vulnerability detection. It emphasizes a holistic evaluation of system defenses and provides actionable insights that strengthen cybersecurity resilience. The significance of this scope can be outlined as follows:

◇ Academic Significance

- Demonstrates the practical application of **Python, Machine Learning, and AI** in cybersecurity.
- Provides a structured framework for students and researchers to study **security posture assessment**.
- Serves as a reference model for integrating **rule-based checks** with **ML-driven predictions**.

◇ Practical Significance

- Enables organizations to **quickly assess defenses** without requiring deep technical expertise.
- Reduces dependency on manual audits by offering **automated, repeatable, and scalable assessments**.
- Provides **real-time recommendations** that can be directly implemented to improve security posture.
- Helps in identifying gaps in **Antivirus, Endpoint Protection, Firewall, and Email Security** before attackers exploit them.

◇ Industrial Significance

- Supports compliance with **CIS Controls, NIST Cybersecurity Framework, and ISO 27001**.
- Assists IT teams in **prioritizing remediation efforts** based on defense scores.
- Can be integrated into **enterprise SOC workflows** for continuous monitoring.
- Enhances **incident readiness** by proactively identifying weak configurations.

◇ Social Significance

- Promotes **cyber hygiene awareness** among individuals and small businesses.
- Provides accessible tools for non-technical users to understand their **security exposure**.
- Contributes to reducing risks of **data breaches, phishing attacks, and malware infections**.

◇ **Scope Boundaries**

- Focused on **IP-based system evaluation** (endpoint, network, and email security).
- Does not replace full-scale penetration testing but complements it with **automated posture checks**.
- Future enhancements may extend scope to **IoT, cloud workloads, and automated remediation**.

CONCLUSION

The proposed **Automated Security Posture Analyzer (ASPA)** aims to evaluate system defenses and determine their effectiveness against potential hacker threats using Python, Machine Learning, and AI techniques. By applying structured scanning, feature extraction, and supervised learning approaches, the project seeks to classify system configurations into relevant categories such as secure, partially secure, or vulnerable. The methodology outlined in this proposal provides a systematic plan for developing an effective and reliable security posture assessment system.

This study highlights the importance of understanding and continuously monitoring cybersecurity defenses in modern digital environments and demonstrates how computational techniques can support better interpretation of system readiness. The work also establishes a foundation for further enhancement during implementation, including advanced modeling, integration with enterprise tools, and performance evaluation. Overall, the project has strong academic and practical relevance, contributing to the broader field of **AI-driven cybersecurity assessment and defense automation**.

Key achievements include:

- Automated scanning of multiple defense layers.
- ML-based classification of secure vs insecure configurations.
- Actionable recommendations mapped to real-world security practices.
- A user-friendly dashboard for visualization and reporting.

◇ Future Scope

To further enhance ASPA, the following improvements can be considered:

1. **Advanced Predictive Modeling**
    - Integrate deep learning models and ensemble techniques to improve accuracy in risk classification.

2. **Real-Time Monitoring**

    - Extend the tool to continuously monitor system defenses and alert users when configurations drift from secure baselines.

3. **IoT and Cloud Integration**

    - Expand coverage to include IoT devices, cloud workloads, and containerized environments, which are increasingly targeted by attackers.

4. **Automated Remediation**

    - Develop scripts or playbooks that not only detect weaknesses but also automatically fix common misconfigurations (e.g., enabling firewall, updating AV signatures).

5. **SIEM and SOC Integration**

    - Connect ASPA with Security Information and Event Management (SIEM) tools or Security Operations Centers (SOC) for enterprise-scale deployment.

REFERENCES

1. **NIST Cybersecurity Framework – Assessment & Auditing Resources.** National Institute of Standards and Technology (NIST). Available at: https://www.nist.gov/cyberframework/assessment-auditing-resources

2. **Assessing Risk and Security Posture with CIS Controls Tools.** Center for Internet Security (CIS). Available at: https://www.cisecurity.org/insights/blog/assessing-risk-and-security-posture-with-cis-controls-tools

3. **Cybersecurity Posture Assessment Tool – GitHub Repository.** cristoxdxd, *Cybersecurity-Posture-Assessment-Tool*. Available at: https://github.com/cristoxdxd/Cybersecurity-Posture-Assessment-Tool

4. **Scikit-learn: Machine Learning in Python.** Pedregosa et al., Journal of Machine Learning Research, 12, pp. 2825–2830, 2011.

5. **psutil Documentation.** Python System and Process Utilities. Available at: https://psutil.readthedocs.io

6. **python-nmap Documentation.** Python wrapper for Nmap. Available at: https://pypi.org/project/python-nmap

7. **dnspython Documentation.** DNS toolkit for Python. Available at: https://www.dnspython.org

8. **Streamlit Documentation.** Streamlit Inc. Available at: https://docs.streamlit.io