CYBERSECURITY AND SECURITY ANALYSIS

Automated Detection of Security Misconfigurations Using Intelligent Tools

M. Akshata, Rabiya Samreen, P. Rachana, Sayeema Iram, R. Reshma

Department of Artificial Intelligence and Data Science, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India

Published October 2026

Abstract

Background: Security misconfigurations in web applications, configuration files, and network services can lead to security risks and exposure of sensitive information. Manually identifying these issues can be time-consuming, especially when there are various types of inputs to check. This paper introduces a web-based system that helps in automating the detection of potential security misconfigurations by using intelligent tools. Users can upload files, enter web links, or provide network port information for analysis. The system processes the submitted input, scans for possible security issues, and categorizes the findings into Low, Medium, and High risk levels. It also offers explanations, recommendations, and corrective actions to help users understand and resolve the identified issues. The system is built using Python and Flask, along with tools and libraries such as Nmap, Requests, Pandas, Scikit-learn, and OpenCV. The main goal is to simplify security analysis and reduce the effort required for manual checks.

Methods: The proposed system accepts configuration files, web URLs, and network port information for analysis. The submitted inputs are validated and processed using Python and Flask. Security scanning tools and predefined rules are used to identify potential misconfigurations. Detected issues are categorized into Low, Medium, and High risk levels, and recommendations along with possible corrective actions are provided to help users address them.

Results: The system is designed to identify potential security misconfigurations in uploaded files, web URLs, and network port information. It aims to present the detected issues in a clear and organized manner by categorizing them into Low, Medium, and High risk levels. The system also provides explanations, security recommendations, and possible corrective actions to assist users in understanding the identified issues. By combining different security checks into a single web interface, the system aims to simplify the analysis process and reduce the effort needed for manual checking. Further testing is required to evaluate its detection accuracy and overall performance.

Conclusion: The proposed system offers a common platform for analyzing potential security misconfigurations using intelligent tools. It integrates input validation, security scanning, risk classification, and recommendation generation into a single workflow. The system aims to help users identify possible security weaknesses and understand how to address them. Its web-based interface is designed to make security analysis more accessible and easier to manage. Future improvements could include adding more detection rules, supporting additional configuration types, and evaluating performance using appropriate test datasets.

1 Introduction

In recent years, the use of web applications, computer networks, and online services has grown rapidly. These technologies have become an essential part of everyday life, business operations, and communication. However, as the use of digital systems continues to expand, ensuring their security has become a major concern. One of the most common security issues faced by these systems is mis-

configuration. Security misconfiguration occurs when applications, servers, network services, or other system components are set up with incorrect, incomplete, or unsafe settings. These mistakes may expose sensitive data, allow unauthorized access, and create opportunities for attackers to exploit system vulnerabilities.

Detecting security misconfigurations is not always straightforward, as different systems have varying configuration requirements and security standards. Some issues might not be easily noticeable during normal system usage, but they can still pose serious security threats. Manually checking configuration files, website security settings, and network services can be time-consuming and requires a certain level of technical expertise. Moreover, interpreting the output of various security tools may be challenging for users unfamiliar with cybersecurity concepts. For these reasons, an automated approach can help simplify the initial security assessment process and make the detection of possible security issues more efficient.

This project focuses on developing a web-based system for detecting potential security misconfigurations using intelligent tools and automated analysis techniques. The system enables users to submit configuration files, website URLs, and network-related information for a security assessment. Based on the input provided, the system reviews the available data, looks for common security issues, and finds potential misconfigurations. The identified problems are then grouped into Low, Medium, and High risk categories to help users understand how serious each issue is and to decide which ones need immediate attention. Along with the risk levels, the system offers appropriate suggestions and helps users enhance their security settings.

The main goal of this project is to combine various security assessment activities into one easy-to-use web interface. Instead of requiring users to perform each check by hand, the system simplifies the process by integrating security scanning, risk evaluation, and recommendation creation. It also presents the findings in a more straightforward way, helping users spot possible weaknesses and take the right steps. Overall, the project is designed to cut down on manual work, increase the efficiency of initial security checks, and aid in making better decisions when dealing with security misconfigurations. The system is meant to assist in spotting common configuration-related risks and should be used as a helpful tool rather than a substitute for a full security audit.

2 Methods

2.1 Problem statement

Security misconfigurations in websites, configuration files, and network services can introduce security risks and leave systems open to attacks. These issues can arise due to improper settings, unnecessary open ports, weak access controls, or outdated security practices. If not identified in a timely manner, they can result in unauthorized access, data breaches, and other security incidents. Manually finding these weaknesses can be time-consuming, particularly when different tools are needed for different types of security checks. To tackle this issue, the project develops an Automated Detection of Security Misconfigurations Using Intelligent Tools. The application brings together configuration file analysis, website security scanning, and network scanning into a single web-based platform. It also provides risk classifications, security reports, and scan results.

2.2 Study design

The system is designed to perform different security checks through a single web application. It provides separate options for scanning configuration files, checking website URLs, and examining network targets. Users can select the required option and provide the necessary input through the interface. The application then sends the submitted information to the appropriate scanning module. After the scan is completed, the available results are displayed on the screen. The overall process includes input validation, scan execution, result analysis, risk classification, and report generation. Each module performs a specific task, making the application easier to understand and manage. At the same time, all modules work within the same platform, so users do not need to switch between different interfaces to review their results. This design also makes it easier to maintain the application and add more security checks in the future.

2.3 Data collection

The system collects the information provided by users through the web interface. This includes configuration files, website URLs, network targets, and port ranges. Each type of input is sent to the module responsible for handling that particular scan. Before processing begins, the application checks whether the required information has been provided in the expected format. Configuration files are submitted for file analysis, website URLs are used for website scanning, and network targets are passed to the network scanning module. The system then collects the information returned by the scanning tools. Depending on the scan, this information may include insecure configuration settings, website security observations, detected open ports, service details, and possible security risks. The collected results are arranged in a structured format so that they can be displayed clearly on the dashboard and individual result pages. This process helps users review the findings without having to manually organize information from different scanning tools.

2.4 Data analysis

The data analysis process begins after collecting information from configuration files, website URLs, and network targets. The system processes each input through the appropriate scanning module to identify potential security weaknesses. Configuration files are examined for insecure settings, websites are checked using the implemented security checks, and network targets are scanned using Nmap to identify open ports and available services. After scanning, the results are collected, organized, and classified into Critical, High, Medium, and Low categories based on the rules implemented in the application. This classification helps users understand the severity of reported issues and prioritize those that require immediate attention. The system presents the findings through the dashboard and detailed result pages, making the information easier to review and understand.

The dashboard provides an overview of scan activity and the distribution of recorded risk categories. Users can examine individual scan results to understand the identified issues and review available supporting evidence. The report generation module presents the findings along with possible recommendations for resolving security weaknesses. These recommendations help users identify suitable corrective measures, such as reviewing configuration settings, restricting unnecessary network access, and applying relevant security updates.

The scan history module maintains access to previously recorded scans, allowing users to review earlier findings and statuses. Comparing similar scans can help users understand changes in the results after configuration updates or other corrective actions. The collected information also

supports organized record management and simplifies the review of previous assessments. Overall, the data analysis process converts scanning outputs into structured information that users can understand and act upon. It brings the results from different scanning modules together within a single platform and supports the identification and prioritization of potential security issues. However, the accuracy of the findings depends on the scanning tools, implemented security rules, and conditions under which each scan is performed. Further testing with known security cases is necessary to evaluate the detection accuracy and reliability of the system.

3 Results

The implemented Automated Detection of Security Misconfigurations Using Intelligent Tools provides a web-based platform for analyzing potential security weaknesses in configuration files, websites, and network services. The system integrates configuration file scanning, website security scanning, and network scanning with a dashboard, risk classification, security report generation, and scan history.

The system provides separate interfaces for submitting configuration files, entering website URLs, and specifying network targets and port ranges. The submitted inputs are processed by the corresponding scanning modules, and the resulting information is displayed through the web interface. The dashboard presents an overview of scan activity and risk classifications. In the displayed dashboard, 18 total scans are recorded, comprising 8 Critical, 5 High, 4 Medium, and 1 Low. These counts should be verified against the stored scan records before being included as final experimental results. The network scanning module uses Nmap to examine the specified target and port range. In one displayed scan result, no open ports were detected within the scanned range. This finding applies only to the particular scan and does not establish that the target is free from all security vulnerabilities.

The security report module presents scan findings, severity information, evidence, and recommendations. The scan history module allows users to review previously recorded details and statuses. These features help users understand the scan results, examine potential security issues, and identify possible corrective actions.

3.1 Configuration file scanning

The configuration scanning module examines uploaded files using the implemented security checks. It helps identify potentially insecure settings and highlights areas that may require further investigation or correction. The module checks the file contents against predefined security rules to identify common configuration mistakes. This helps users understand the possible risks associated with their configuration settings and take suitable corrective measures. The module also presents the detected issues in an organized manner, allowing users to review the findings more easily. It helps users recognize settings that may need to be changed to improve system security. By simplifying the process of examining configuration files, the module reduces the effort required for manual inspection and helps users identify potential security weaknesses at an early stage.

3.2 Website security scanning

The website scanning module accepts website URLs and performs the supported security checks. The results help users review potential website security issues and identify areas that may need additional attention. The module checks for the presence of selected security headers and identifies

those that may be missing from the website's response. This information helps users understand possible weaknesses and improve the security configuration of their websites.

3.3 Network scanning

The network scanning module uses Nmap to examine the specified target and port range. In one displayed scan, no open ports were detected within the scanned range. This result applies only to that particular scan and does not confirm that the target is completely secure. The module helps users understand which network ports are accessible and identify services that may need further investigation. The collected scan information can be used as a starting point for reviewing network exposure and identifying potential security concerns.

3.4 Dashboard and risk classification

The dashboard displays an overview of scan activity and risk categories. The displayed dashboard records 18 total scans: 8 Critical, 5 High, 4 Medium, and 1 Low. These counts should be verified against the stored records before final submission. The risk classification helps users understand the reported severity and prioritize further investigation. The dashboard presents the scan information in an organized manner, making it easier to review the overall security status. It also helps users identify higher-risk findings that may require more immediate attention.

3.5 Security reports and scan history

The report generation module presents available findings, severity levels, evidence, and recommendations. The scan history feature allows users to review previously recorded scans and their statuses. These functions help organize assessment results and make earlier findings easier to access. The reports provide useful information about the issues detected during scanning and suggest possible steps to address them. Maintaining scan history also helps users revisit earlier results and keep track of previous security assessments.

Table 1. Functional testing results.

Test Case	Expected Result	Outcome
Configuration File Scanning	Analyzes uploaded files	Successful
Website Security Scanning	Processes website URLs	Successful
Network Scanning	Displays scan information	Verified
Risk Classification	Displays severity levels	As Expected
Dashboard	Displays scan summaries	Verified
Report Generation	Presents findings and recommendations	Generated
Scan History	Displays previous scan records	Retrieved

3.6 Overall results

Overall, the implemented system brings configuration file scanning, website security scanning, and network scanning together in a single web-based platform. The dashboard provides a summary of scan activity and risk classifications, helping users understand the recorded results. The scanning modules process their respective inputs and display the available findings through the application interface. The report generation feature presents security observations, severity levels, and

recommendations, while the scan history allows users to review previously recorded assessments. These features make the security assessment process more organized and easier to manage. The system helps users identify potential security weaknesses and determine which issues require further investigation. Although the application supports preliminary security analysis, its effectiveness depends on the implemented scanning rules and the accuracy of the results returned by the tools. Further testing with verified security cases can help evaluate its detection accuracy, reliability, and overall performance. The system provides a foundation for improving security practices and reducing the effort required to identify potential Misconfiguration.

4 Discussion

The Automated Detection of Security Misconfigurations Using Intelligent Tools provides a common platform for identifying and reviewing potential security weaknesses in configuration files, websites, and network services. The system combines different scanning modules with a dashboard, risk classification, report generation, and scan history. This integration makes the security assessment process more organized and allows users to access the available findings through a single web application.

The configuration file scanning module helps examine uploaded files for potentially insecure settings. Incorrect configuration parameters can create security weaknesses and affect the protection of an application or system. By examining supported files, the module helps users identify settings that may require further investigation or correction. The effectiveness of this process depends on the file formats supported and the security rules implemented in the application. Adding more configuration checks could improve the coverage of this module.

The website security scanning module allows users to submit website URLs and perform the security checks supported by the system. The findings help users review potential website security issues and understand which areas may require further attention. Website security can be affected by several factors, including incorrect settings, insufficient protection, and outdated components. However, the application can identify only the issues covered by its implemented checks. Therefore, the results should be considered preliminary findings rather than a complete assessment of website security.

The network scanning module uses Nmap to examine specified targets and port ranges. This helps users understand which ports or services are detected during a scan. In one displayed result, no open ports were detected within the selected range. This observation applies only to that particular scan and does not confirm that the target is completely secure. Network accessibility, filtering rules, and scan settings may influence the results. Testing additional authorized targets and port ranges could provide a better understanding of the module's behaviour. Risk classification helps organize the findings according to their assigned severity levels, such as Critical, High, Medium, and Low. This makes it easier for users to understand the relative importance of reported issues and prioritize further investigation. Higher-severity findings may require earlier attention, while other findings can be reviewed according to their context and possible impact. However, the accuracy of the classification depends on the rules implemented in the application. The severity level should therefore be considered together with the available evidence and the conditions of the affected system.

The dashboard provides an overview of recorded scan activity and the distribution of risk categories. It allows users to review the available information without opening every scan individually. This improves the presentation of results and helps users understand the overall scanning activity. The dashboard counts should match the stored scan records to ensure consistency. It is also important to

distinguish the number of scans from the number of individual findings, since a single scan may identify multiple issues.

The security report generation feature presents the available findings in a structured format. Reports may contain scan details, severity information, supporting evidence, and recommendations, depending on the information returned by the scanning modules. This helps users understand the reported issues and identify possible corrective actions. Recommendations may include reviewing configuration settings, restricting unnecessary network access, and applying appropriate security updates. Clear and well-organized reports also make it easier to refer to findings during further investigation.

The scan history module allows users to review previously recorded scans and their available details. This feature is useful when assessments need to be repeated after configuration changes or corrective actions. By examining comparable records, users can identify changes in scan results and investigate whether previously reported issues remain. Maintaining accurate scan history also supports organized record management and makes earlier findings easier to access.

Although the system combines several useful functions, its effectiveness depends on the scanning tools, implemented security checks, and quality of the submitted inputs. Some security weaknesses may remain undetected if they fall outside the scope of the available checks. Similarly, a scan that returns no findings does not guarantee that the target is free from security vulnerabilities. The application should therefore be treated as a supporting tool for preliminary security assessment rather than a replacement for comprehensive security testing.

Further improvements could include adding more security checks, supporting additional configuration file formats, improving the explanation of findings, and providing more detailed recommendations. Testing with known security cases would help evaluate the accuracy and reliability of the scanning modules. Performance measurements, including processing time, detection accuracy, and false-positive rates, could also provide a clearer understanding of the system's capabilities. Such improvements would help identify limitations and guide future development.

Overall, the system demonstrates how configuration file scanning, website security scanning, and network scanning can be integrated into a single web-based platform. The dashboard, risk classification, reporting, and history features help users review potential security weaknesses in an organized manner. With further testing and improvements, the application could become a more effective tool for supporting security assessments and helping users identify possible Misconfiguration.

5 Conclusion

The Automated Detection of Security Misconfigurations Using Intelligent Tools provides a web-based platform for identifying potential security weaknesses in configuration files, websites, and network services. The integration of scanning modules, risk classification, dashboard, report generation, and scan history makes the assessment process more organized and easier to manage. The system helps users review findings, understand their severity, and identify possible corrective actions. With additional security checks and proper testing, the application can be further improved to support more reliable security assessments.

6 Recommendations

1. Improve scanning accuracy: Add more security rules and test them using known security cases.
2. Expand file support: Support additional configuration formats to identify more potential weaknesses.
3. Enhance website scanning: Include more security checks to improve the coverage of website assessments.
4. Improve reporting: Provide clear explanations, evidence, and recommendations for each finding.
5. Strengthen validation: Test the application with authorized targets to evaluate its accuracy and reliability.
6. Future development: Add more scanning features and improve the user interface for easier security assessment.

References

[1] OWASP Foundation. *OWASP Top 10: Security Misconfiguration* [Internet]. Available from: https://owasp.org/Top10/

[2] Johnson A, Dempsey K, Ross R, Gupta S, Bailey D. *Guide for Security-Focused Configuration Management of Information Systems*. NIST Special Publication 800-128. Gaithersburg, MD: NIST. https://doi.org/10.6028/NIST.SP.800-128

[3] Lyon G. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, CA: Insecure.Com LLC; 2009. Available from: https://nmap.org/book/

[4] OWASP Foundation. *Web Security Testing Guide* [Internet]. Available from: https://owasp.org/www-project-web-security-testing-guide/

[5] OWASP Foundation. *OWASP Top Ten Web Application Security Risks* [Internet]. Available from: https://owasp.org/www-project-top-ten/

[6] Center for Internet Security. *CIS Critical Security Controls v8.1* [Internet]. Available from: https://www.cisecurity.org/controls/cis-controls-list

[7] National Institute of Standards and Technology. *Cybersecurity Framework (CSF) 2.0* [Internet]. Available from: https://www.nist.gov/cyberframework

[8] National Institute of Standards and Technology. *Security and Privacy Controls for Information Systems and Organizations*. NIST SP 800-53, Revision 5. Gaithersburg, MD: NIST; 2020. Available from: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

[9] National Institute of Standards and Technology. *National Vulnerability Database* [Internet]. Available from: https://nvd.nist.gov/

[10] OWASP Foundation. *OWASP Application Security Verification Standard* [Internet]. Available from: https://owasp.org/www-project-application-security-verification-standard/

[11] MITRE Corporation. *Common Vulnerabilities and Exposures (CVE)* [Internet]. Available from: https://www.cve.org/

[12] MITRE Corporation. *Common Weakness Enumeration (CWE)* [Internet]. Available from: https://cwe.mitre.org/

[13] Python Software Foundation. *Python Documentation* [Internet]. Available from: https://docs.python.org/3/

[14] Flask Documentation. *Flask Web Development Framework* [Internet]. Available from: https://flask.palletsprojects.com/

[15] OWASP Foundation. *OWASP Cheat Sheet Series* [Internet]. Available from: https://cheatsheetseries.owasp.org/