Vulnerability scanner that automates the detection of common security flaws and provides AI-driven insights for impact assessment and remediation suggestions

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

***Abstract*— As the complexity and volume of software applications continue to grow, so too does the risk of security vulnerabilities. Traditional vulnerability scanners, while effective at identifying basic flaws, often lack the intelligence to assess the true impact of vulnerabilities or offer tailored remediation advice (Shin et al., 2015). This research introduces an advanced vulnerability scanning system that not only automates the detection of common security flaws such as SQL injection, Cross-Site Scripting (XSS), insecure deserialization, and misconfigured security headers but also leverages Artificial Intelligence (AI) to evaluate the impact and suggest context-aware remediation strategies (Almukaynizi et al., 2021). By integrating machine learning models trained on historical exploit data and secure coding practices, the system prioritizes vulnerabilities based on severity and environment-specific parameters (Sabottke et al., 2015). The proposed solution aims to bridge the gap between automated scanning and intelligent threat analysis, reducing false positives and enhancing overall application security posture**

Keywords—Security, Vulnerability, Machine Learning, Artificial Intelligence, Scanner, Natural Language Processing,

I. INTRODUCTION

In the digital era, web and software applications are essential to every industry. However, they also introduce attack vectors that malicious actors can exploit. According to the Verizon Data Breach Investigations Report (2023), vulnerabilities in applications remain a significant cause of data breaches across industries. As software development lifecycles accelerate with Agile and DevOps practices, the need for efficient, automated, and intelligent vulnerability assessment tools has become more pressing (Chen et al., 2019).

Traditional vulnerability scanners—both commercial and open-source—have limitations in their ability to adapt to diverse environments or provide customized advice. They typically produce static reports without considering the runtime context of the application, which hinders prioritization and effective remediation (Avgerinos et al., 2014). Furthermore, the abundance of false positives leads to alert fatigue among developers and security professionals (Sabottke et al., 2015).

Artificial Intelligence (AI) and Machine Learning (ML) provide promising solutions to these challenges. By learning from historical vulnerabilities, threat patterns, and developer behaviors, AI can provide more accurate and context-aware insights. When integrated with vulnerability scanners, AI can not only improve detection but also assess risk dynamically and provide meaningful, environment-specific remediation guidance (Shar et al., 2013). This paper proposes a hybrid solution that combines rule-based scanning with AI-driven impact analysis to offer comprehensive security evaluation and support timely mitigation efforts

II. LITERATURE REVIEW

Traditional scanners such as Nessus, Nikto, OpenVAS, and OWASP ZAP are widely used to identify known vulnerabilities through signature-based detection, static rules, and pre-defined payloads (Scarfone & Mell, 2007).

These tools have been successful in detecting basic web vulnerabilities, configuration errors, and missing patches. However, they are primarily limited to known CVEs and often fail to detect zero-day threats or complex chained vulnerabilities (Shin et al., 2015).

Static Application Security Testing (SAST) tools such as SonarQube and Checkmarx analyze source code for potential vulnerabilities without executing it. They provide early detection during development but struggle with runtime-specific flaws (Zalewski, 2012). On the other hand, Dynamic Application Security Testing (DAST) tools like Burp Suite test live applications during execution, enabling detection of injection attacks and logic flaws, but can be time-consuming and hard to integrate into CI/CD pipelines (Avgerinos et al., 2014).

AI is increasingly used in cybersecurity to detect anomalies, forecast attacks, and assess risks. Research by Hu et al. (2021) demonstrated that deep learning models can identify vulnerabilities by analyzing code semantics rather than relying on rule-based logic. Similarly, machine learning techniques have been applied to bug and vulnerability prediction using features like code complexity, developer activity, and commit history (Shar et al., 2013)..

DeepCode, CodeQL, and Semgrep represent a new generation of tools leveraging AI to understand code intent and detect subtle issues missed by traditional tools. However, their impact assessment capabilities are still evolving and often lack integration with live context or real-time threat intelligence (Almukaynizi et al., 2021).

Despite significant advancements, most current vulnerability scanners are siloed. They either focus on detection or offer generalized remediation suggestions. Few integrate AI to assess real-time business impact or consider environment-specific factors such as application architecture, data sensitivity, or user roles (Sabottke et al., 2015). Additionally, existing literature lacks models that integrate both technical severity and exploit likelihood into a unified risk score—an essential feature for prioritization in real-world scenarios (Chen et al., 2019).

This research aims to fill these gaps by proposing a hybrid AI-driven vulnerability scanner that enhances traditional detection techniques with intelligent impact analysis and contextualized remediation suggestions.

III. PROPOSED METHODOLGY

The proposed methodology

IV. DATASET

Dataset Information

V. DATA LOADING AND EXPLORATION

The

VI. DATA PREPROCESSING

Data preprocessing

A. Handling missing values:

The presence of missing values

B. Feature scaling

Numerical

C. Encoding categorical variables

Categorical

D. Splitting the dataset:

The dataset is

VI. MODEL TRAINING

A. MODEL 1

B. MODEL 2

VII. PERFORMANCE EVALUATION

Performance Evaluation

A. EVALUATION METRICS

Evaluation Metrics

VI. PERFORMANCE COMPARISION

Performance Comparison

VIII. RESULTS AND DISCUSSION

Results and Discussion

IX. FUTURE DIRECTIONS

Future Directions

X. REFERENCES

[1] Almukaynizi, M., Pastrana, S., Mohaisen, A. (2021). A Machine Learning-Based Framework for Vulnerability Detection in Source Code. IEEE Access, 9, 84364–84378..

[2] Avgerinos, T., Cha, S. K., Hao, B. L., & Brumley, D. (2014). AEG: Automatic Exploit Generation. Communications of the ACM, 57(2), 74-84

[3] Chen, H., Li, Y., Guo, Y., & Zhu, Y. (2019). A Survey of Security Vulnerability Detection, Exploitation, and Patch Generation. ACM Computing Surveys (CSUR), 52(4), 1-36.

[4] Sabottke, C., Suciu, O., & Dumitras, T. (2015). Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. USENIX Security Symposium

..