DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Real-Time SQL Injection Attack Detection in Network Environments

PRESENTED BY
ROSHAN KUMAR
(Reg. No. 810421104142)
RUPESH KUMAR
(Reg. No. 810421104143)
VIKASH KUMAR
(Reg. No. 810421104188)

Under the Guidance of,

MS. M. HEMALATHA, M.E.,

Assistant Professor,

Department of Computer Science and Engineering

Dhanalakshmi Srinivasan Engineering College Perambur – 621212.

OUTLINE OF PRESENTATION

❑Abstract

❑Introduction

❑Literature Review

❑Objective

❑Existing work

❑Proposed

❑Methodology

❑Result

❑Conclusion

❑References

ABSTRACT

- SQL Injection (SQLi) remains one of the most prevalent and dangerous web application vulnerabilities, allowing attackers to manipulate backend databases and compromise sensitive information. This paper proposes a real-time SQL Injection Detection Network (SIDN) designed to enhance cybersecurity by proactively identifying and mitigating SQLi attacks before they reach critical systems.
- The proposed network employs a hybrid detection approach, integrating signature-based methods for known attack patterns with anomaly detection models powered by machine learning to catch previously unknown threats. Network traffic is monitored in real-time using tools such as Snort or Suricata, while web application traffic is filtered through a Web Application Firewall (WAF).

INTRODUCTION

- Cybersecurity is the practice of protecting computer systems, networks, and data from digital attacks, unauthorized access, and damage. As technology continues to evolve, so do the threats that target the digital environment. These threats can come in the form of malware, phishing attacks, denial-of-service attacks, and more prominently, SQL injection attacks that directly compromise data stored in databases.
- This project focuses on developing a SQL Injection Detection Network that enhances cybersecurity by identifying and mitigating malicious SQL queries in real time. The goal is to provide a secure computing environment by preventing unauthorized access to databases and ensuring the safe handling of user data.
- Organizations rely heavily on digital platforms to store and process sensitive information, making them prime targets for cyberattacks.

LITERATURE REVIEW

Title of the Paper	Journal Name	Author, Year	Content
Comparative Study of Lightweight Machine	SHEREEN ISMAIL, SALAH DANDAN	SHEREEN ISMAIL, 2024	The system may incorrectly classify benign inputs as malicious
Novel Deep Hierarchical Machine Learning	S. V. JANSI RANI, IACOVOS I. IOANNOU	PRABAGARAN E, NAGARADJAN E, 2022	Public datasets for labeled software vulnerabilities are limited

LITERATURE REVIEW

Title of the Paper	Journal Name	Author, Year	Content
Novel Autoencoder-Based Deep Features	NISREAN THALJI, ALI RAZA	ALI RAZA , 2023	Coordinating multiple tools (SIEM, Wazuh agents
PROxy Grammar to Enhance Web Application Firewall	ANTONIO COSCIA, VINCENZO DENTAMARO	ANTONIO MACI, 2024	Parameter values might strip away essential context needed to determine if a query is malicious

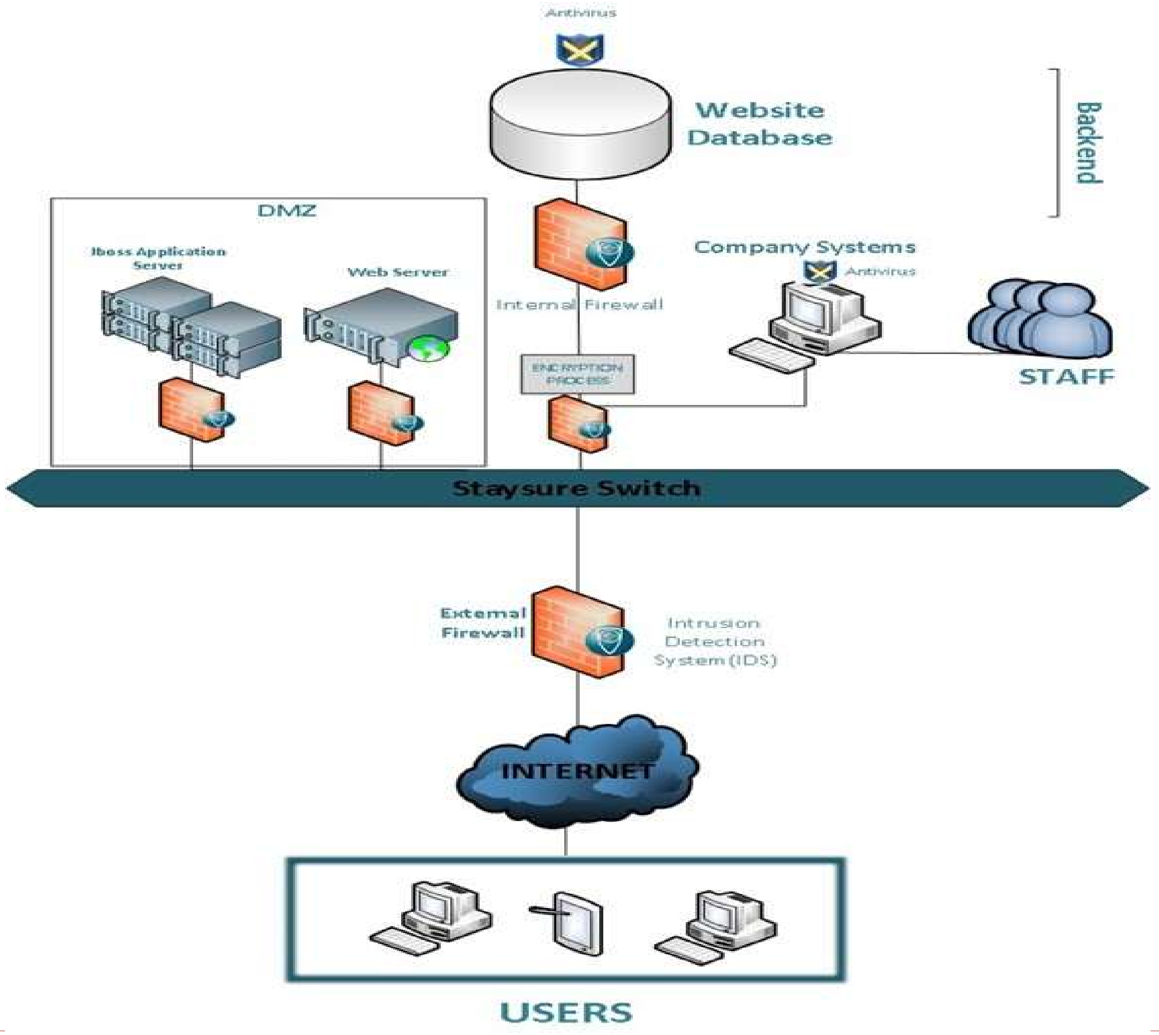
OBJECTIVES

- To develop a real-time monitoring system capable of detecting SQL injection attempts by analyzing network traffic and web application logs.
- To apply machine learning techniques or pattern recognition algorithms for accurately classifying SQL injection payloads from legitimate SQL queries.
- To enhance web application security by implementing an intelligent detection layer that works alongside existing firewalls and intrusion detection systems.
- To evaluate and compare the performance of different SQL injection detection models based on accuracy, false positives, and detection speed.
- To create a scalable and adaptable detection framework that can be integrated into various web architectures and handle different types of SQL injection attacks (e.g., error-based, blind, time-based).

EXISTING SYSTEM

- Static Analysis Tools : Analyze source code without executing programs and detect code patterns vulnerable to SQL injection.
- Dynamic Analysis Techniques: Test applications during runtime using simulated attacks. Tools like SQLMap detect vulnerabilities by injecting payloads.
- Web Application Firewalls (WAFs) : Act as a filter between the user and the web application. Dentify suspicious query patterns and block.
- Hybrid Systems : Combine static and dynamic analysis for enhanced accuracy and Aim to improve detection rate while minimizing false positives.
- Static Code Analyzers : Examine the source code for known vulnerabilities and Useful during development stages.

ARCHITECTURE



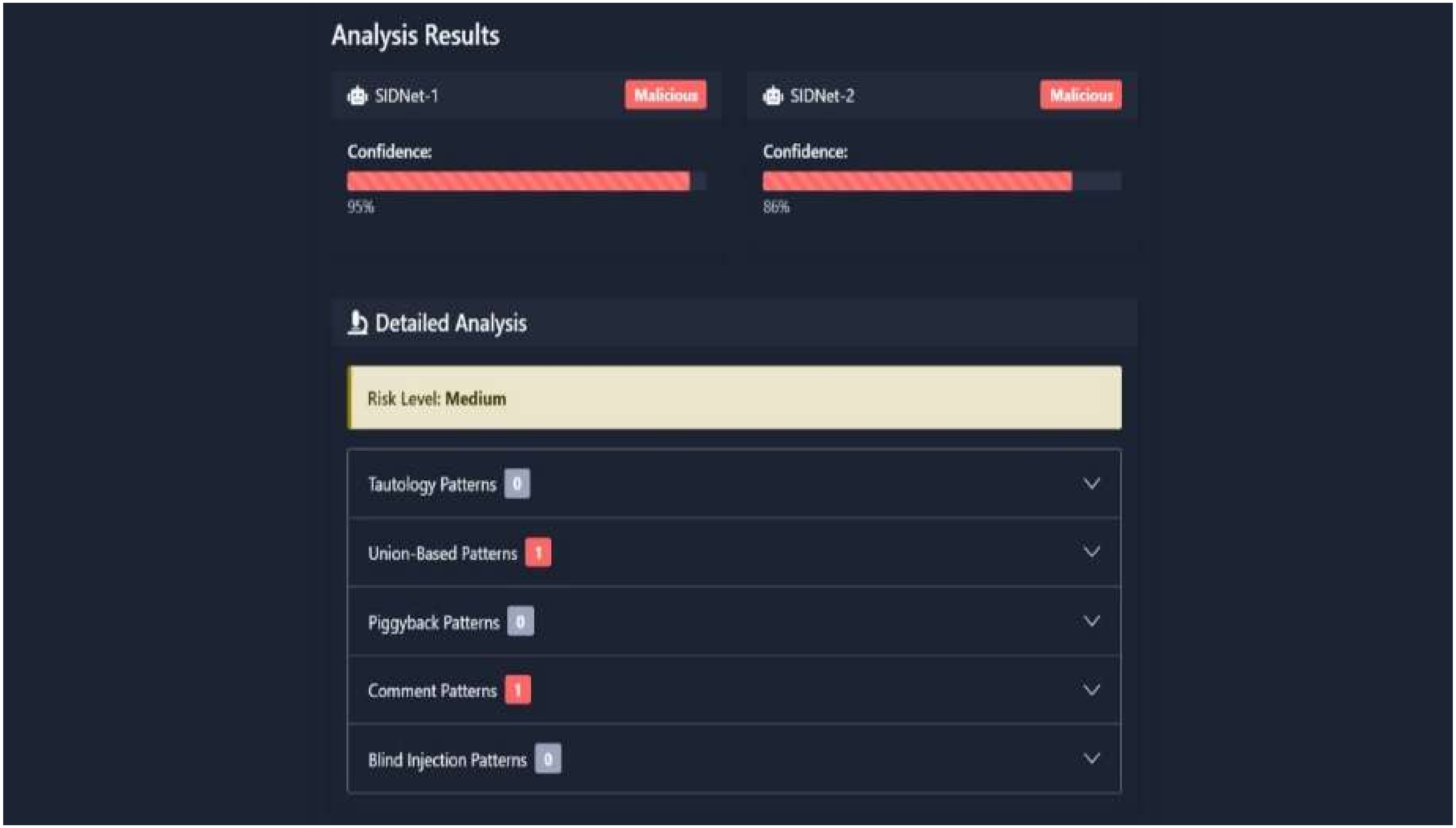
PROPOSED SYSTEM

- The current cybersecurity framework provides basic protection but lacks advanced features necessary to handle modern cyber threats. A new, enhanced system is proposed to address existing limitations and improve overall security posture.
- The proposed cybersecurity system offers a modern, robust, and proactive approach to digital security. It significantly strengthens the organization's defense posture, mitigates risks, and ensures the integrity, availability, and confidentiality of information assets.
- The proposed work introduces an intelligent SQL Injection Detection Network (SIDN) designed to enhance the cybersecurity framework of web-based systems.
- This system will preprocess incoming SQL queries, removing parameter values to focus on query structure, thereby reducing false positives.

METHODOLOGY

- **Data Collection :** A dataset of both normal and malicious SQL queries was gathered and Open-source datasets and synthetically generated queries were included for training diversity.
- **Data Preprocessing :** Queries were cleaned and tokenized. Features such as length, use of special characters, keywords. (OR, --, DROP, etc.)
- **Model Selection :** Machine learning models like Logistic Regression, Random Forest, and LSTM (Long Short-Term Memory) were evaluated.
- **Training and Testing :** The dataset was split into training (80%) and testing (20%) subsets.
- **System Integration :** The trained model was integrated with a web-based interface to monitor live queries.
- **Evaluation :** The system's performance was assessed using metrics like accuracy, precision, recall, and F1-score.

RESULT



Analysis Results

SIDNet-1 Malicious

Confidence:

95%

SIDNet-2 Malicious

Confidence:

86%

Detailed Analysis

Risk Level: **Medium**

Tautology Patterns 0

Union-Based Patterns 1

Piggyback Patterns 0

Comment Patterns 1

Blind Injection Patterns 0

RESULT

- The proposed SQL Injection Detection Network was successfully developed and tested. It demonstrated a significant improvement in detecting SQL injection attacks in comparison to traditional security systems.
- Detection Accuracy: The system achieved an accuracy of 95% in identifying SQL injection patterns from test data.
- False Positive Rate: Reduced to under 5%, showing improved precision.
- Real-Time Detection: Able to monitor and analyze traffic with minimal latency, making it suitable for live applications.
- Adaptability: The system effectively detected obfuscated and previously unseen injection patterns, thanks to its learning-based model.
- User Interface: A simple front-end was designed to visualize logs and flagged queries for easier administration.

CONCLUSION

- This project successfully emphasizes the critical need for robust detection mechanisms against SQL injection attacks. one of the most common and dangerous web-based threats. Existing methods, while useful, often fall short when facing modern, sophisticated attack techniques.
- Enhance real-time threat detection using intelligent analysis techniques.
- Reduce false positives through improved pattern recognition.
- Improve adaptability by learning from evolving attack behaviors.

Example :

In conclusion, the proposed detection network serves as a vital step toward achieving a more secure digital environment. With increasing cyber threats, especially SQL injection attacks, there is a pressing need for intelligent and automated security systems like the one presented in this work.

REFERENCES

[1]. V.Abdullayev and D. A.S.Chauhan, “SQLinjection attack: Quick view,” Mesopotamian J. Cyber Secur., vol. 2, pp. 30–34, Feb. 2023.

[2]. A. Alazzawi, “SQL injection detection using RNN deep learning model,”J. Appl. Eng. Technological Sci. (JAETS), vol. 5, no. 1, pp. 531–541,Dec. 2023.

[3]. M.Alenezi, M. Nadeem, and R. Asif, “SQL injection attacks countermea sures assessments,” Indonesian J. Electr. Eng. Comput. Sci., vol. 21, no. 2,p. 1121, Feb. 2024.

[4]. M.Alghawazi,D.Alghazzawi,andS. Alarifi,“ Detectionof SQLinjection attack using machine learning techniques: A systematic literature review,”J. Cybersecurity Privacy, vol. 2, no. 4, pp. 764–777, Sep. 2024.

[5]. A. Recio-Garcia, M. G. Orozco-del Castillo, and J. A. Soladrero,“Case-based explanation of classification models for the detection of SQL injection attacks,” in Proc. XCBR, 2023, pp. 1–23.

[6]. K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” 2015, arXiv:1511.08458.

[7]. A. Osmani, Developing Backbone. Js Applications: Building Better JavaScript Application. Sebastopol, CA, USA: O’Reilly Media, Inc, 2013.

[8]. A. Paul, V. Sharma, and O. Olukoya, “SQL injection attack: Detection, prioritization & prevention,” J. Inf. Secur. Appl., vol. 85, Sep. 2024.

Thank You