PROJECT REPORT ON

Malicious URL & Payload Inspector

Submitted by

LINGAMALLU VASU DEVA SIVA SAI SUBBARAO

Table of Contents

1. Introduction

The Malicious URL & Payload Inspector is a web-based cybersecurity tool built using Streamlit that allows users to analyze the safety of URLs. The system uses machine learning models (Random Forest & SVM) trained on lexical features and flags common SQL injection and XSS attack patterns.

This tool is intended for cybersecurity professionals, developers, and end-users to pre-validate URLs for threats before interacting with them, enhancing internet safety and reducing exposure to phishing or injection-based attacks.

2. Existing System

Existing tools like VirusTotal, PhishTank, or command-line scanners rely heavily on API calls, signature-based detection, or large datasets. They:

- Often require internet access and third-party APIs.
- May not detect obfuscated or zero-day URLs.
- Do not provide local ML-based predictions or source-level inspection.
- Lack a GUI or are only available as browser extensions.

3. Proposed System

The proposed system addresses the above limitations by offering:

- A lightweight, standalone solution with no dependency on external APIs.
- Support for custom dataset training and label encoding.
- Real-time detection of SQLi and XSS payloads embedded in URLs.
- Two classification models (Random Forest and SVM) with interpretable predictions.
- Dual input modes: Single URL and CSV upload for bulk analysis.
- Fully responsive Streamlit GUI with export options.

4. System Design

4.1 Architecture

- Input Layer: Accepts URL string(s) via GUI.
- Processing Layer:
  - Extracts lexical features.
  - Applies SQLi/XSS detection patterns.
  - Runs predictions using ML models.
- Output Layer:
  - Displays prediction results.
  - Allows result download as CSV.

4.2 Components

- Frontend: Streamlit UI with CSS styling.
- Backend:
  - Feature extraction module.
  - SQLi/XSS detection via regex.
  - ML prediction using Random Forest & SVM.
- Model Loader: joblib-based loader for models and label encoders.

4.3 Workflow

1. User inputs URL(s).
2. Features extracted via regex and parsing.
3. Classification models output predictions.
4. SQLi/XSS signatures are flagged.
5. Results are displayed or exported.

4.4 Technologies Used

- Python 3

- Libraries: Streamlit, joblib, scikit-learn, pandas, re, urllib

- Models: Random Forest, SVM (trained on lexical URL features)

- Deployment: Web-based Streamlit App

5. Implementation

5.1 Module Description

Feature Extraction Module: Extracts lexical indicators such as URL length, IP presence, suspicious keywords, digit/letter count, etc.

Detection Module: Regex patterns detect SQL Injection attempts and XSS payloads.

ML Prediction Module: Uses Random Forest and SVM to classify URL as malicious or benign.

User Interface Module: Streamlit-based GUI offers single URL scan, CSV upload, real-time display and CSV export.

5.2 System Flow

User Input (URL or CSV)

->

Feature Extraction (Regex, Parsing)

->

SQLi & XSS Detection

->

ML Prediction (RF + SVM)

->

Result Display + Export Option

6. Algorithm Implementation

- Feature Extraction: O(1) per URL

- Prediction: O(1)

- SQLi/XSS Detection: O(K)

- Total Time Complexity: O(N) for N URLs

- Space Complexity: O(N)

7. Requirement Specification

7.1 Functional Requirements

- Analyze single or multiple URLs

- Detect suspicious patterns

- Classify with ML models

- Display and export results

7.2 Software Requirements

- Python 3.7+

- Libraries: streamlit, joblib, pandas, sklearn, re

- Models: rf_model.pkl, svm_model.pkl, label_encoder.pkl

7.3 Hardware Requirements

- CPU: Dual-core or better

- RAM: 4 GB minimum

- Storage: 100 MB

- Internet: Not required

8. Testing

Test Types: Unit Testing, Integration Testing, Functional Testing, Black Box Testing

Results: All modules passed tests, Model predictions verified, Streamlit UI functions correctly

9. Conclusion

The Malicious URL & Payload Inspector effectively identifies phishing and payload-injected URLs. With lexical feature extraction, dual-model classification (RF and SVM), and detection of SQLi/XSS threats, the tool provides a practical solution for real-time URL inspection. The GUI makes it accessible for all user levels, and CSV support enables scalable testing. The project meets cybersecurity needs while maintaining simplicity and speed.