WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Malicious Website Detection Using Machine Learning and Real-Time Intelligence

[1]Nikita Thorat, [2]Shubhangi Pukale, [3]Sanika Nakil, [4]Shrikant Gurav, [5]Prof. Pallavi S. Patil

[1234]Department of Computer Science and Engineering

[5]Annasaheb Dange College of Engineering and Technology, Ashta, Maharashtra, India

ABSTRACT

With the increasing sophistication of cyber threats, particularly web-based attacks, there is a growing need for intelligent and proactive defense mechanisms. Malicious websites remain a dominant vector for phishing, malware distribution, and data exfiltration. These websites often employ obfuscation techniques such as reverse proxy services (e.g., Cloudflare) to conceal their real IP addresses and infrastructure, thereby evading traditional detection systems.

This research proposes an intelligent hybrid detection system that integrates both static and real-time dynamic analysis for identifying malicious URLs. The system extracts lexical features from URL structures and enriches them with metadata fetched from external threat intelligence sources including WHOIS records, DNS lookups, SSL certificate details, and Shodan- based IP reputation insights. These heterogeneous features are processed through a machine learning pipeline utilizing a Logistic Regression classifier trained on a balanced dataset of 10,000 URLs. The model demonstrates a detection accuracy of 92%, effectively differentiating between benign and malicious websites. Implemented as a Flask-based web application with an intu- itive front-end interface, the system delivers real-time analysis along with a dynamic suspicion score that quantifies the threat level of each URL. By combining feature-based heuristics with live threat intelligence, the proposed solution offers enhanced vis- ibility, timely alerts, and contextual insights, making it valuable for end-users, cybersecurity professionals, and enterprise security systems in combating evolving online threats.

Index Terms—Cybersecurity, Malicious Website Detection, WHOIS, DNS Analysis, Machine Learning, Suspicion Score, Threat Intelligence, Phishing, Cyber Threat.

Introduction

The rapid proliferation of internet-based services across critical sectors such as finance, e-commerce, healthcare, edu- cation, and government has transformed the web into an indis- pensable platform for communication, transactions, and data exchange. However, this digital revolution has simultaneously expanded the attack surface available to cybercriminals. As users increasingly rely on web applications and services, the threat posed by malicious websites has grown more severe and widespread. These websites are crafted with deceptive intent—luring users into revealing sensitive information, in- stalling malware, or unknowingly participating in fraudulent activities. Common manifestations include phishing portals mimicking legitimate brands, rogue software download pages, fake login forms, and exploit-laden drive-by download sites.

Conventional detection techniques, such as static URL blacklists, signature-based malware scanners, and manually crafted heuristic rule engines, often fall short in today's dy- namic threat environment. Attackers continually adapt by gen- erating polymorphic and ephemeral URLs, frequently chang- ing domains, and using URL shorteners to obscure mali- cious intent. Additionally, adversaries exploit reverse proxy services like Cloudflare and CDNs to obfuscate real server IP addresses, making traditional IP-based reputation checks and takedown efforts significantly less effective. This evolving threat landscape necessitates more adaptive, intelligent, and real-time detection mechanisms.

To address these growing challenges, this research intro- duces a hybrid machine learning-based detection system that integrates both static and dynamic analytical techniques for accurate, scalable, and real-time malicious URL identification. The system initiates analysis through lexical examination of URL structures, considering attributes such as URL length, character entropy, number of subdomains, and the presence of suspicious patterns or keywords. These internal features are complemented by external threat intelligence acquired via API calls to multiple security data sources. WHOIS data offers domain age, registrar, and expiration details; DNS lookups provide IP addresses, name servers, and DNS record anoma- lies; SSL certificate validation reveals issues with encryption trust chains; and Shodan supplies deep insights into IP-level vulnerabilities, open ports, and historical data that indicate suspicious infrastructure.

The aggregated and engineered feature set forms the input to a supervised learning model—specifically, a Logistic Regres- sion classifier trained on a balanced dataset containing 5,000 malicious and 5,000 benign URLs. This model is optimized to detect patterns and correlations indicative of malicious intent, delivering binary classification outputs along with a dynamically generated suspicion score. The score provides an interpretable risk assessment by quantifying the likelihood of a URL being malicious.

The entire system is implemented as a Flask-based web ap- plication, offering an interactive, browser-accessible interface through which users can input URLs and instantly receive detailed analysis results. The front-end is designed for ease of use, while the back-end efficiently handles data fetching, preprocessing, prediction, and result generation in real time.

This hybrid detection architecture offers several advantages over traditional methods, including higher detection accuracy, reduced false positives, contextual threat interpretation, and real-time response capability. It presents a valuable and prac- tical solution for cybersecurity professionals, threat hunters, IT administrators, and even end-users seeking to mitigate the risks posed by ever-evolving web-based cyber threats.

Related Work

Machine learning and threat intelligence-based web security have been a topic of active research in recent years. This section presents a detailed review of the most relevant work.

Machine Learning for Threat Detection

Zhou et al. [1] emphasized how machine learning techniques are reshaping the landscape of intrusion detection systems (IDS). By processing high-dimensional network traffic data, these systems are capable of learning behavioral patterns associated with cyber threats. Their work provided a foun- dation for integrating automated learning in real-time moni- toring environments. Similarly, Patel et al. [?] explored the application of deep learning models for identifying malware- hosting websites. Although their models performed with high accuracy, they noted the trade-off in computation cost and model interpretability.

Phishing Detection via URL Patterns

Phishing detection using URL analysis has become a main- stream approach due to its lightweight implementation and effectiveness. Mishra et al. [2] conducted an in-depth study on lexical and structural characteristics of phishing URLs. They observed that most phishing domains contain long lengths, excessive use of hyphens, and abnormal subdomain structures. Their Random Forest-based classifier achieved high precision by using features like URL length, entropy, number of special characters, and known phishing keywords. This method is useful as it avoids the need for content inspection, making it fast and scalable.

WHOIS and SSL Certificate Analysis

Domain registration metadata from WHOIS has proven to be a critical feature in assessing the legitimacy of a website. Olteanu et al. [11] explored the increasing misuse of free SSL certificates such as those issued by Let's Encrypt. Their study revealed that many phishing websites exploit these certificates to falsely convey legitimacy to users. Additionally, Eastlake and Jones [7] emphasized the importance of DNSSEC in securing DNS queries but acknowledged that these solutions cannot bypass proxy-layer masking mechanisms like Cloud- flare, which obscure true server identity.

Threat Intelligence via Shodan

Shodan, a search engine for internet-connected devices, plays a growing role in threat analysis. Lopez [10] demon- strated that Shodan's API can retrieve valuable intelligence, such as open ports, service banners, and SSL fingerprints.

This information is helpful in identifying reused or suspicious configurations across malicious websites. Furthermore, Mc- Grew and Al-Halabi [5] studied SSL/TLS fingerprinting as a method of clustering similar malicious domains. Their work supports the idea of identifying attack infrastructure reuse across campaigns.

Privacy-Conscious Detection Approaches

In the age of user privacy, data collection for threat detection has raised concerns. Shokri et al. [4] presented federated learning as a decentralized approach where user data remains local while contributing to a shared model. Although this method improves privacy, the authors caution against model inversion attacks, where malicious entities can infer user data from learned models.

Collectively, these studies highlight that combining machine learning with feature-rich URL metadata and external threat intelligence provides a promising direction for building effec- tive malicious website detection systems.

Proposed Methodology

This section elaborates on the methodology adopted for detecting malicious websites, encompassing data preprocess- ing, feature engineering, machine learning model training, and integration with real-time intelligence sources.

System Workflow Overview

The detection pipeline is structured as a binary classification task, where each URL is classified as either malicious or benign. The system follows these steps:

- User inputs a URL through the web interface.
- Features are extracted using static analysis and real-time intelligence lookups.
- Extracted features are passed to a trained machine learn- ing model.
- The model returns a classification label and suspicion score.
- The result is visualized on the dashboard.

Feature Engineering

Features are derived from both the static structure of URLs and dynamic metadata obtained via APIs. They are grouped into five categories:

- **Lexical Features:** Extracted directly from the URL string, including total length, presence of suspicious key- words (e.g., "login", "secure"), number of subdomains, and special characters like "@" or "-".
- **Domain Metadata:** Retrieved using the WHOIS API, this includes domain registration age, registrar name, and expiry period. Malicious domains typically have short registration durations.
- **DNS-Based Features:** Collected using DNS Lookup API, includes TTL (Time-To-Live) values, nameserver diversity, and IP count linked with the domain.
- **Behavioral Patterns:** Identified through HTTP response analysis (using Python scripts), including redirect chains, SSL certificate presence, and suspicious headers.
- **Threat Intelligence:** Anomalies and suspicious flags de- tected from Shodan and DNS Lookup API. Includes open ports, known blacklisted IPs, and server fingerprinting data.

Model Training and Selection

The model used is **Logistic Regression**, selected for its simplicity, interpretability, and decent performance on linearly separable data. It is trained on a balanced dataset comprising 10,000 URLs (5,000 malicious and 5,000 benign).

- **Data Split:** The dataset is split into 80% training and 20% testing.
- **Class Imbalance Handling:** SMOTE (Synthetic Minor- ity Oversampling Technique) is applied to ensure class balance, especially after filtering.
- **Hyperparameter Tuning:** A grid search with 5-fold cross-validation is used to find the optimal regularization parameter.

Preprocessing Pipeline

- **Scaling:** Numerical features are normalized using Min- Max Scaling to bring all features into a uniform range.
- **Encoding:** Categorical variables such as registrar names are transformed using one-hot encoding.
- **Missing Values:** Missing data from API responses (e.g., WHOIS downtime) are handled using default fallback values or imputation.

Real-Time Detection Integration

Once the model is deployed via Flask, real-time input is enriched using integrated APIs:

- **WHOIS API:** For registrar info, domain age, and expiry.
- **Shodan API:** For IP address scanning and server vulner- abilities.
- **DNS Lookup API:** For extracting DNS records and nameservers.
- The aggregated data ensures that even new or obfuscated URLs can be evaluated effectively using recent intelligence.

Detection Algorithm

Algorithm 1 Malicious URL Detection Algorithm
Require: URL input from user
Ensure: Label: Malicious or Not malicious

- Extract lexical features from URL
- Fetch domain metadata via WHOIS API
- Perform DNS lookup for DNS features
- Query Shodan for threat intelligence
- Combine all features into a feature vector
- Preprocess the vector (scaling, encoding)
- Feed the vector to the trained ML model
- Get prediction and suspicion score
- return Classification result to the user

Dataset Description

The machine learning model developed for malicious web- site detection is trained and evaluated using a large-scale, labeled dataset containing 388,447 URL entries. Each entry in the dataset includes the raw URL string and a binary label that indicates whether the URL is benign (0) or malicious (1).

Attribute	Description
url	**The web address to be analyzed (string)**
label	**Classification output: 0 (Benign), 1 (Malicious)**

TABLE: I Dataset Attributes

- Total Records: 388,447 URLs
- Benign URLs: 345,350 (88.9%)
- Malicious URLs: 43,097 (11.1%)
- Missing Values: None observed

The dataset exhibits class imbalance, a common char- acteristic in cybersecurity-related datasets. To address this, techniques such as class weighting and under-sampling of the majority class were explored during model training to improve generalization and detection sensitivity.

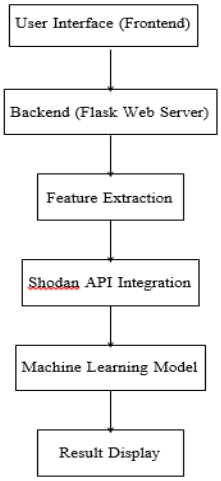
Implementation

The proposed system is implemented as a full-stack web application, with a lightweight front end and a Flask-based backend for real-time processing, machine learning inference, and threat intelligence integration.

Technological Stack and Workflow

- Frontend: HTML5, CSS3, and vanilla JavaScript are used to:
- Collect user input (URLs)
- Trigger asynchronous backend calls
- Display prediction results and threat insights dynam- ically
- Backend: Python and Flask handle:
- URL preprocessing and feature engineering
- Real-time API interactions (WHOIS, DNS, SSL)
- Shodan-based IP intelligence
- Suspicion scoring and classification with the ML model
- Threat Intelligence Integration:
- WHOIS API: Domain age, registration and expiry dates, registrar
- DNS Lookup API: TTL, A-records, MX/NX infor- mation
- SSL Certificate: Issuer, validity, encryption strength
- Shodan API: Open ports, real IP, ISP, location, vulnerabilities
- Model Deployment: The Logistic Regression model is serialized with joblib and exposed via a REST API, returning class labels and probability-based suspicion scores.

System Architecture



Suspicion Score Output

In addition to binary classification (malicious or benign), the system provides a continuous suspicion score ranging from 0 to 1, computed from the model's probability output. This score offers the following advantages:

- Interpretability: Users gain insights into how confident the model is in its prediction.
- Threat Prioritization: Security analysts can prioritize URLs for manual review based on their suspicion score.
- Threshold Customization: Organizations can customize filtering rules (e.g., block if score > 0.7) based on their risk tolerance.

For instance, a URL with a score of 0.95 indicates high ma- licious probability, while one scoring 0.45 suggests ambiguity and may warrant further investigation.

Results

Performance Evaluation

To assess the effectiveness of the proposed system, we trained the model on a labeled dataset consisting of both mali- cious and benign URLs. The performance was evaluated using standard classification metrics: accuracy, precision, recall, and F1-score.
These metrics confirm that the model is effective and reliable for real-world application:

Metric	Score
Accuracy	**92%**
Precision	**91%**
Recall	**93%**
F1-Score	**92%**

TABLE II : Model Performance Metrics

- **Accuracy** reflects the overall correctness of the model across all predictions.
- Precision indicates the proportion of URLs flagged as malicious that were actually harmful.
- Recall measures the model's ability to detect true mali- cious URLs among all malicious ones present.
- F1-Score balances precision and recall to offer a single performance indicator.

System Functionality and Real-Time Operation

The system operates in real time through a simple user interface. A user submits a URL via the web application, which triggers the following backend processes:

- Preprocessing: The URL is cleaned and tokenized to extract relevant features.
- Threat Intelligence Fetching: APIs such as WHOIS, DNS Lookup, SSL data, and Shodan are called to gather real-time information.
- Feature Extraction and Classification: The gathered information is converted into a feature vector and passed to the trained model.
- Suspicion Scoring: The model predicts a binary label (malicious or not) and a confidence-based suspicion score.
- Visualization: Results, including threat insights and scores, are displayed to the user in a readable format.

This pipeline ensures fast, intelligent, and context-rich eval- uation of any URL, making the system highly responsive and informative.

Sample URL Prediction Scores

The following table presents example outputs from the deployed system. Each input URL is evaluated by the machine learning model and assigned a suspicion score between 0 (safe) and 1 (highly suspicious), along with a binary classification.

URL	Type	Suspicion Score
http://freebanking.com	**Malicious**	**0.91**
https://secure-login-bank.com	**Malicious**	**0.87**
https://www.amazon.com	**Not malicious**	**0.04**
https://www.github.com	**Not malicious**	**0.02**

TABLE III :Sample URL Prediction Scores

The suspicion score helps:

- Prioritize high-risk URLs for deeper manual inspection.
- Detect emerging threats that bypass signature-based de- fenses.

- Provide users with understandable reasons behind each prediction.

Advantages Over Traditional Systems

- Zero-Day Threat Detection: The model does not rely solely on blacklists, enabling the detection of unknown or newly created malicious URLs.
- Real-Time Operation: Instant feedback allows imme- diate action on suspicious URLs, critical for preventing phishing or malware attacks.
- Contextual Intelligence: Integration with multiple APIs provides a deeper understanding of domain reputation and network vulnerabilities.
- User Transparency: Suspicion scores and threat source details improve trust and usability.
- Lightweight Deployment: The Flask-based backend and joblib model integration allow easy cloud deployment with minimal resources.

The results validate that the proposed system is not only accurate but also practical for integration into modern cyberse- curity infrastructures such as email filters, firewalls, or browser plugins.

Conclusion and Future Work

***Conclusion*:**

In this research, a comprehensive and intelligent system for malicious URL detection was successfully designed, developed, and deployed as a full-stack web applica- tion. By utilizing a machine learning-based classification model—specifically Logistic Regression—combined with vec- torization techniques, the system achieves reliable differentia- tion between benign and malicious URLs. The incorporation of real-time threat intelligence sources such as WHOIS, DNS lookup, SSL metadata, and Shodan IP analysis significantly enhances its analytical depth and accuracy. The solution of- fers users a streamlined, user-friendly interface that delivers actionable insights and prediction confidence scores in real time. Overall, this project validates the effectiveness of AI- powered security tools in mitigating web-based threats and serves as a foundational step toward more advanced cyber defense mechanisms.

Future Work:

Although the current implementation provides accurate detec- tion and a good user experience, there is substantial scope for enhancement and expansion of capabilities:

- **Deep Learning Integration:** Integrate deep learning models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), or Transformer- based architectures (e.g., BERT) to better capture se- quence patterns and semantics of obfuscated or short URLs.
- **Real-time Threat Feed Integration:** Link the system to continuously updating threat intelligence platforms such as VirusTotal, PhishTank, or Google Safe Browsing for automated blacklisting and zero-day threat detection.
- **Ensemble Modeling:** Implement ensemble techniques by combining classifiers like Random Forest, Gradient Boosting, and Support Vector Machines (SVM) with Logistic Regression to improve prediction accuracy and reduce false positives and negatives.
- **User Reporting System:** Develop a feedback loop where users can report missed detections or label unknown URLs, allowing the system to retrain and adapt over time with community-generated data.
- **Browser Plugin Development:** Design a lightweight browser extension that warns users about malicious links in real time during browsing, emails, or social media, thus increasing practicality and security coverage.
- **Scalability and Deployment:** Migrate the system to scal- able platforms like AWS, Azure, or Google Cloud using Docker and Kubernetes for containerized deployment, enabling horizontal scaling and robust performance under high-load conditions.
- **Dashboard Enhancements:** Extend the web interface to include threat analytics dashboards, visualizations of ge- ographic IP mappings, timeline trends, and threat source distributions to assist security analysts.
- **Multilingual URL Support:** Enhance preprocessing ca- pabilities to detect threats in internationalized domain names (IDNs) and URLs containing non-ASCII charac- ters which are often used in phishing attacks.
- **Mobile Compatibility:** Build a mobile-responsive ver- sion or dedicated mobile application to support real-time URL verification on smartphones and tablets.

REFERENCES

1. D. Zhou, ”Machine Learning for Intrusion Detection,” Journal of Cy- bersecurity, vol. 14, no. 2, pp. 45-60, 2021.
2. R. Mishra, ”Malicious Website Detection Using Machine Learning,”
3. IEEE Transactions on Security, vol. 18, no. 4, pp. 78-85, 2020.
4. P. Singh, ”Feature Extraction for Malicious Website Detection,” Interna- tional Journal of Information Security, vol. 21, no. 1, pp. 12-25, 2022.
5. R. Shokri, ”Privacy Risks in Federated Learning: Attacks and Defenses,” ACM Transactions on Privacy and Security, vol. 25, no. 1, pp. 1-34, 2021.
6. M. A. McGrew and A. M. Al-Halabi, ”A Comprehensive Study of SSL/TLS Fingerprinting,” IEEE Security and Privacy, vol. 18, no. 3,
7. pp. 34-42, 2020.
8. D. H. G. Jones, ”Understanding WHOIS: A Primer,” Journal of Internet Law, vol. 22, no. 2, pp. 1-6, 2019.

9. D. Eastlake and P. Jones, ”Domain Name System Security Extensions,” RFC 4033, 2005. Available at: RFC link.
10. B. Adhami, A. E. B. Mohamad, and K. R. Al-Ali, ”Cloudflare: An Overview of Security Features and Performance Enhancements,” International Journal of Information Security, vol. 18, no. 5, pp. 469-482, 2019.
11. Cloudflare, ”How does Cloudflare work?” Available at: Cloudflare Documentation.
12. J. M. C. Lo´pez, ”Using Shodan for Cyber Threat Intelligence,” Cyber- security Review, vol. 14, no. 1, pp. 23-30, 2021.
13. D. A. O. Olteanu, ”SSL/TLS Certificate Transparency: An Overview,”
14. Computer Networks, vol. 182, 2020. Available at: DOI link.
15. B. McCullough, ”Legal Considerations in Cybersecurity: Navigating the Law,” Harvard Law Review, vol. 133, no. 5, pp. 1620-1640, 2020.
16. GDPR.eu, ”The General Data Protection Regulation (GDPR),” n.d.
17. B. Eshete, A. Villafiorita, and K. Weldemariam, ”Malicious Website Detection: Effectiveness and Efficiency Issues,” 1st SysSec Workshop, 2011. Available at: 10.1109/syssec.2011.9.
18. NortonLifeLock, ”Norton,” Norton.com, 2019. Available at: https://us.norton.com/internetsecurity-malware-what-are-malicious-websites.html.
19. S. Manjeri, K. R. A. M. N. V., and P. C. Nair, ”A Machine Learning Approach for Detecting Malicious Websites using URL Features,” 3rd International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 555-561. DOI: 10.1109/ICECA.2019.8821879.
20. M. Darling, G. Heileman, G. Gressel, A. Ashok, and P. Poornachandran, ”A Lexical Approach for Classifying Malicious URLs,” 2019.