PhishGator: Detection of Password-Stealing Phishing Attacks Using HTML Template Analysis

Arshita Sharma
University of Tennessee, Knoxville

Devanshi Patel
University of Tennessee, Knoxville

Sai Deepika Dasari
University of Tennessee, Knoxville

Abstract

Phishing attacks remain a prevalent cybersecurity threat, leveraging deceptive websites to steal sensitive user credentials such as passwords [6]. These attacks pose significant risks to both individuals and organizations, often resulting in financial losses and compromised data integrity. *PhishGator*, a comprehensive machine learning-based system, was developed to address these challenges. It focuses on detecting phishing websites and analyzing credential-stealing behaviors embedded in HTML templates.

Using the Gradient Boosting Classifier (GBC), a model known for its robust handling of non-linear feature interactions, *PhishGator* achieves an impressive detection accuracy of approximately 97%. The system analyzes 32 features extracted from URLs and HTML content, such as the presence of IP addresses, HTTPS absence, and the use of obfuscating symbols like @. Additionally, *PhishGator* employs advanced mechanisms to detect credential-stealing behaviors, including password fields, domain mismatches, non-secure data submission protocols, and keylogging scripts.

This report delves into the methodology, implementation, and performance evaluation of *PhishGator*, emphasizing its dual-layer approach. The combination of phishing detection and credential-theft analysis allows for a more granular classification of threats, categorizing websites as phishing, credential-stealing, or both. Comparative evaluations with other models highlight the superior performance of the GBC.

PhishGator represents a critical advancement in the fight against phishing, with potential applications in browser extensions, corporate email systems, and user training modules. While the results are promising, challenges such as dynamic content analysis, false positives, and evolving phishing tactics underscore the need for ongoing research and enhancements. This study provides insights into the integration of machine learning in cybersecurity, offering a scalable and effective solution to mitigate phishing threats.

1 Introduction

Phishing attacks have become one of the most pervasive cybersecurity threats in recent years, targeting individuals, organizations, and governments worldwide [6]. These attacks exploit user trust by imitating legitimate websites and online services, leading unsuspecting victims to disclose sensitive information such as usernames, passwords, and financial data. The global costs associated with phishing extend beyond financial loss, affecting organizational reputation, customer trust, and overall cybersecurity resilience [5].

Despite significant advancements in cybersecurity [2], traditional phishing detection mechanisms often fall short. Static rule-based systems are frequently circumvented by sophisticated attackers employing dynamic URL generation, obfuscated HTML, and socially engineered content. These limitations demand robust, adaptable solutions that leverage advanced technologies like machine learning to identify and mitigate phishing threats.

In this report, we introduce *PhishGator*, a comprehensive system designed to detect phishing websites and identify credential-stealing behaviors. By combining phishing detection with advanced HTML template analysis, *PhishGator* addresses both traditional phishing threats and emerging methods of credential theft.

1.1 Problem Statement

Phishing continues to pose critical challenges in the field of cybersecurity, evolving in both sophistication and scale [2]. Traditional detection methods, such as blacklist-based approaches, are inherently reactive and limited in scope, often proving inadequate in addressing the novel techniques deployed by attackers [1]. This issue is further exacerbated by the lack of robust mechanisms to detect credential-stealing behaviors, such as keylogging scripts and insecure form submissions. The primary challenges include:

Dynamic Phishing Techniques: Modern attackers frequently rely on dynamic URLs, obfuscated content, and so-

cial engineering tactics to evade conventional detection systems [6].

Credential Theft: Legitimate-looking websites often employ malicious mechanisms that exploit user trust to steal sensitive credentials, creating an additional layer of vulnerability.

False Positives: Many detection systems generate a high rate of false positives, flagging legitimate websites as malicious, which undermines user trust in these technologies.

Addressing these pressing challenges requires a holistic and integrated approach that combines phishing detection and credential-theft analysis into a single, scalable framework.

1.2 Objectives

The *PhishGator* project is driven by several key objectives aimed at overcoming the limitations of existing phishing detection systems:

Developing a robust detection system: A machine learning-based system capable of accurately detecting phishing websites while maintaining a low false-positive rate, aligned with current cybersecurity framework standards [5].

Incorporating HTML analysis: Utilizing advanced HTML template analysis to detect credential-stealing behaviors such as insecure forms, domain mismatches, and keylogging scripts.

Comprehensive classification: Providing a system capable of distinguishing between safe websites, phishing websites, credential-stealing websites, and those exhibiting both threats.

Evaluating model performance: Assessing the effectiveness of the Gradient Boosting Classifier (GBC) in handling the complexities of phishing detection and comparing it with alternative machine learning models.

1.3 Motivation

The urgency of protecting users and organizations from the financial and reputational damage caused by phishing attacks forms the foundation of this research [6]. Current detection mechanisms often fail to keep pace with the rapidly evolving nature of phishing tactics, leaving users vulnerable to increasingly sophisticated threats.

Machine learning offers an innovative and scalable solution by enabling the analysis of complex, non-linear relationships between features extracted from URLs and HTML content. *PhishGator* leverages these capabilities to set a new standard in phishing detection, creating a robust and adaptive tool for combating phishing and credential theft.

The potential applications of this research are extensive, ranging from browser extensions and corporate security tools to educational programs aimed at raising awareness. By addressing the gaps in existing solutions [2], *PhishGator* contributes to the broader effort to enhance cybersecurity resilience and protect digital ecosystems from the persistent threat of phishing.

2 Related Work

The detection of phishing attacks and credential-theft behaviors has been a topic of extensive research within the cybersecurity domain [2]. Previous studies have explored various techniques and methodologies, ranging from heuristic-based approaches to sophisticated machine learning algorithms. This section provides an overview of the existing body of work in phishing detection, credential-theft analysis, and the application of machine learning in addressing these challenges.

2.1 Previous Phishing Detection Techniques

Phishing detection has traditionally relied on static and heuristic-based methods, which include blacklist and whitelist mechanisms [5]. Blacklist-based approaches maintain a repository of known phishing URLs and block access to them. However, this approach is inherently reactive and limited in its ability to handle newly emerging phishing websites. Whitelist approaches, on the other hand, restrict access to a predefined list of trusted domains, but they often prove impractical for general users due to their restrictive nature.

Other studies have utilized rule-based systems, where characteristics such as URL length, the presence of special characters, or missing HTTPS protocols are flagged as potential indicators of phishing. While effective to some extent, these systems suffer from a high rate of false positives and lack adaptability against dynamic phishing tactics [1].

In recent years, content-based detection mechanisms have gained traction [4]. These methods analyze the HTML and JavaScript content of websites to identify suspicious patterns. Techniques such as DOM tree comparison, hidden redirections, and analysis of embedded resources have shown promise. However, they often require significant computational resources and are vulnerable to obfuscation techniques employed by attackers.

subsectionCredential-Theft Behavior Analysis

Credential theft represents a specific and critical subset of phishing attacks, where malicious actors exploit user trust to harvest sensitive information such as usernames and passwords [6]. Prior work in this area has largely focused on identifying insecure form submissions and domain mismatches [2].

Studies have demonstrated that phishing websites frequently use non-secure protocols (HTTP) or submit form data to external domains unrelated to the website's origin. These behaviors can be detected by analyzing the HTML content of websites, particularly input fields of type `password` and `form` tags.

Another significant area of research has been the detection of keylogging scripts. Attackers often embed JavaScript

snippets that monitor user keystrokes, enabling the capture of sensitive credentials. Methods such as behavior analysis of script execution and heuristic evaluation of JavaScript patterns have been employed to flag such malicious activity [4]. While effective, these methods require advanced parsing and runtime analysis, which can introduce overhead.

2.2 Machine Learning Applications in Phishing

Machine learning has emerged as a transformative tool in phishing detection, enabling systems to adapt and learn from evolving attack patterns [5]. Supervised learning techniques, including Logistic Regression, Support Vector Machines (SVM), Random Forests, and Gradient Boosting, have been widely adopted for this purpose.

Several studies have highlighted the effectiveness of feature-based approaches, where URL attributes such as length, number of subdomains, presence of special characters, and domain age are used as input features for classification models [3]. HTML content features, including external resource loading and form handling mechanisms, have also proven effective in identifying phishing websites.

Gradient Boosting algorithms, in particular, have demonstrated superior performance due to their ability to model complex, non-linear relationships among features. These algorithms excel in handling imbalanced datasets, which is a common challenge in phishing detection where legitimate websites often vastly outnumber phishing sites [7].

Deep learning methods, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have also been explored. These models are particularly effective in detecting phishing URLs and patterns within HTML content. However, their implementation often requires extensive computational resources and large datasets for training.

In summary, the integration of machine learning techniques has significantly advanced phishing detection capabilities [2]. However, gaps remain in effectively addressing credential-theft behaviors and ensuring scalability in real-world applications. The *PhishGator* project builds on these advancements by combining traditional and modern techniques to provide a comprehensive solution to phishing and credential theft.

3 Methods

This section outlines the methodology adopted for designing, implementing, and evaluating the PhishGator system. The methods emphasize the dataset description, feature extraction, machine learning model training, and credential-stealing detection [3].

3.1 Dataset Used

We utilized two datasets for training (`phishing.csv`) and testing (`test_url.csv`). The training data was obtained from Kaggle and consists of 32 features, following established methodologies for cybersecurity research [4]. The test dataset includes unseen URLs from PhishTank and OpenPhish.

3.1.1 phishing.csv

The following describes the 32 features used for training our phishing detection model, aligned with established cybersecurity framework guidelines [6]:

1. **Index**
2. **UsingIP:** Checks if the domain part of the URL is an IP address, such as `http://134.67.9.198/deception.html`.
   Rule: IF {
   The Domain Part has an IP Address → **Phishing**
   Otherwise → **Legitimate**
   }
3. **LongURL:** Examines the URL length.
   Rule: IF {
   URL length < 54 → **Legitimate**
   URL length ≥ 54 and ≤ 75 → **Suspicious**
   Otherwise → **Phishing**
   }
4. **ShortURL:** Identifies shortened URLs (e.g., TinyURL).
   Rule: IF {
   TinyURL → **Phishing**
   Otherwise → **Legitimate**
   }
5. **Symbol@:** Detects the presence of the `@` symbol in the URL.
   Rule: IF {
   URL contains @ symbol → **Phishing**
   Otherwise → **Legitimate**
   }
6. **Redirecting//:** Checks the position of `//` in the URL.
   Rule: IF {
   The position of the last occurrence of "//" > 7 → **Phishing**
   Otherwise → **Legitimate**
   }
7. **PrefixSuffix:** Checks for the presence of - in the domain name.
   Rule: IF {
   Domain name contains "-" symbol → **Phishing**
   Otherwise → **Legitimate**
   }

8. **SubDomains:** Counts the number of dots in the domain and sub-domain.
   Rule: IF {
   Dots in domain part = 1 → **Legitimate**
   Dots in domain part = 2 → **Suspicious**
   Otherwise → **Phishing**
   }

9. **HTTPS:** Evaluates HTTPS presence and certificate trustworthiness.
   Rule: IF {
   Using HTTPS and issuer is trusted and age of certificate ≥ 1 year → **Legitimate**
   Using HTTPS and issuer is not trusted → **Suspicious**
   Otherwise → **Phishing**
   }

10. **DomainRegLen:** Checks domain expiration duration.
    Rule: IF {
    Domain expires in ≤ 1 year → **Phishing**
    Otherwise → **Legitimate**
    }

11. **Favicon:** Identifies if the favicon is loaded from an external domain.
    Rule: IF {
    Favicon loaded from external domain → **Phishing**
    Otherwise → **Legitimate**
    }

12. **NonStdPort:** Checks for use of non-standard ports.
    Rule: IF {
    Port number is not preferred → **Phishing**
    Otherwise → **Legitimate**
    }

13. **HTTPSDomainURL:** Detects the use of HTTP tokens in the domain part of the URL.
    Rule: IF {
    Using HTTP token in domain part of URL → **Phishing**
    Otherwise → **Legitimate**
    }

14. **RequestURL:** Measures the percentage of webpage resources loaded from external domains.
    Rule: IF {
    % of request URL < 22% → **Legitimate**
    % of request URL ≥ 22% and ≤ 61% → **Suspicious**
    Otherwise → **Phishing**
    }

15. **AnchorURL:** Similar to `RequestURL`, focuses on anchor tag URLs.
    Rule: IF {
    % of anchor URLs < 31% → **Legitimate**
    % of anchor URLs ≥ 31% and ≤ 67% → **Suspicious**
    Otherwise → **Phishing**
    }

16. **LinksInScriptTag:** Checks for meta, link, and script tags linked externally.
    Rule: IF {
    % of links in `<Meta>`, `<Script>`, and `<Link>` < 17% → **Legitimate**
    % of links in `<Meta>`, `<Script>`, and `<Link>` ≥ 17% and ≤ 81% → **Suspicious**
    Otherwise → **Phishing**
    }

17. **ServerFormHandler:** Checks if the server form handler refers to a different domain or contains blank information.
    Rule: IF {
    Server form handler is "about: blank" or empty → **Phishing**
    Server form handler refers to a different domain → **Suspicious**
    Otherwise → **Legitimate**
    }

18. **InfoEmail:** Detects use of `mail()` or `mailto:` functions to collect user information.
    Rule: IF {
    Using "mail()" or "mailto:" function to submit user information → **Phishing**
    Otherwise → **Legitimate**
    }

19. **AbnormalURL:** Checks if the URL can be identified via WHOIS database.
    Rule: IF {
    The host name is not included in the URL → **Phishing**
    Otherwise → **Legitimate**
    }

20. **WebsiteForwarding:** Counts the number of times the website redirects.
    Rule: IF {
    Number of redirects ≤ 1 → **Legitimate**
    Number of redirects ≥ 2 and < 4 → **Suspicious**
    Otherwise → **Phishing**
    }

21. **StatusBarCust:** Checks for changes in the status bar on mouse events.
    Rule: IF {
    On mouse over changes status bar → **Phishing**
    Otherwise → **Legitimate**
    }

22. **DisableRightClick:** Identifies if right-click functionality is disabled.
    Rule: IF {

Right click disabled → **Phishing**
Otherwise → **Legitimate**
}

23. **UsingPopUpWindow:** Detects pop-up windows asking for sensitive information.
Rule: IF {
Popup window contains text fields → **Phishing**
Otherwise → **Legitimate**
}

24. **IFrame Redirection:** Checks for the use of `iframe`.
Rule: IF {
Using iframe → **Phishing**
Otherwise → **Legitimate**
}

25. **AgeOfDomain:** Evaluates the age of the domain.
Rule: IF {
Age of domain $\geq$ 6 months → **Legitimate**
Otherwise → **Phishing**
}

26. **DNSRecord:** Verifies the existence of a DNS record for the domain.
Rule: IF {
No DNS Record → **Phishing**
Otherwise → **Legitimate**
}

27. **WebsiteTraffic:** Checks the website's ranking based on traffic.
Rule: IF {
Website Rank $<$ 100,000 → **Legitimate**
Website Rank $>$ 100,000 → **Suspicious**
Otherwise → **Phishing**
}

28. **PageRank:** Measures the importance of the website based on PageRank.
Rule: IF {
PageRank $<$ 0.2 → **Phishing**
Otherwise → **Legitimate**
}

29. **GoogleIndex:** Checks if the website is indexed by Google.
Rule: IF {
Webpage Indexed by Google → **Legitimate**
Otherwise → **Phishing**
}

30. **LinksPointingToPage:** Counts the number of links pointing to the page.
Rule: IF
    - Links = 0 → **Phishing**
    - $0 <$ Links ≤ 2 → **Suspicious**
    - Otherwise → **Legitimate**

31. **StatsReport:** Identifies URLs present in known phishing reports.
Rule: IF {
Host Belongs to Top Phishing Domains/IPs → **Phishing**
Otherwise → **Legitimate**
}

32. **Class:** This is the target variable indicating whether the URL is phishing or legitimate.

3.2 Feature Extraction

Feature extraction aimed to identify critical indicators that differentiate phishing websites from legitimate ones, following established content analysis methodologies [4]. Two primary categories of features were analyzed: **URL-Based Features**: These included metrics like URL length, the presence of IP addresses, and symbols such as "@" and "-" [2]. **HTML-Based Features**: Included properties like insecure form submissions, external resource loading, and script behaviors indicative of phishing attempts [5]. All categorical features were encoded into numerical values, and normalization was applied to standardize feature ranges. This preprocessing improved the performance and reliability of the machine learning model.

3.3 Machine Learning Models Evaluated

Multiple machine learning models were evaluated to classify websites as phishing or legitimate, following rigorous qualitative research standards [7]. After extensive experimentation, the Gradient Boosting Classifier (GBC) was chosen for its robustness in handling complex classification tasks. The evaluation process involved: **Data Splitting**: The dataset was divided into an 80-20 ratio for training and testing, ensuring reliable performance metrics. **Hyperparameter Tuning**: Parameters such as tree depth (set to 4) and learning rate (set to 0.7) were optimized to balance complexity and generalization. **Performance Metrics**: Accuracy, precision, recall, and F1-score were used to measure model effectiveness, achieving a test accuracy of 97%.

3.4 Credential-Stealing Detection Mechanism

To complement phishing detection, a dedicated mechanism was developed to identify credential-theft behaviors by analyzing HTML templates [3]. The mechanism focused on: **Password Fields**: Flagging the presence of password input fields in web forms. **Domain Mismatch**: Detecting form submissions that redirect data to domains other than the originating website. **Non-HTTPS Forms**: Identifying forms that transmit sensitive data over non-secure protocols. **Keylogging**

Scripts: Analyzing JavaScript patterns indicative of keylogging attempts. The results of this analysis were integrated with phishing detection, categorizing websites as safe, phishing, credential-stealing, or both.

4 Results

This section presents the outcomes of the machine learning model evaluations, a comparative analysis of the models, and examples of predictions to highlight the efficacy of the *PhishGator* system, following established research evaluation methodologies [7].

4.1 Model Performance Metrics

The Gradient Boosting Classifier (GBC) demonstrated superior performance in phishing detection, achieving high accuracy, precision, recall, and F1-score, aligning with industry standards for cybersecurity tools [5]. Table 1 summarizes the performance metrics for the model on the test dataset.

Table 1: Model Performance Metrics for Gradient Boosting Classifier

Metric	Value (%)
Accuracy	97
Precision	95
Recall	97
F1-Score	96

These results highlight the GBC's ability to balance precision and recall, ensuring effective classification of phishing and legitimate websites. The model's performance demonstrates its robustness in identifying malicious websites while minimizing false positives and negatives.

4.2 Comparative Analysis of Models

Several machine learning models were evaluated during the study to identify the best performer [4]. Table 2 provides a comparative overview of the models, highlighting the accuracy achieved by each:

Table 2: Comparative Analysis of Machine Learning Models

Model	Accuracy (%)
Logistic Regression	85
Random Forest	89
Support Vector Machine (SVM)	88
Gradient Boosting Classifier	97

Among the evaluated models, the Gradient Boosting Classifier consistently outperformed others, especially in handling the complexities of imbalanced data and feature interactions. Logistic Regression, while efficient, struggled to capture nonlinear relationships. Random Forest showed promise but faced overfitting issues during hyperparameter tuning. Support Vector Machines exhibited competitive accuracy but were computationally expensive, particularly for larger datasets.

These findings underscore the superiority of GBC for phishing detection tasks due to its ability to handle complex feature relationships and imbalanced datasets effectively. Its application ensures high detection accuracy and generalization to unseen data.

4.3 Examples of Predictions

To illustrate the effectiveness of the *PhishGator* system, following established cybersecurity framework guidelines [6], two representative examples of predictions are provided below:

Example 1: Legitimate Website

- **URL:** `https://www.example.com/home`
- **Phishing Detection:** Legitimate
- **Credential-Stealing Detection:** None
- **Final Verdict:** Safe Website

Example 2: Phishing + Credential-Stealing

- **URL:** `http://192.168.1.1/login`
- **Phishing Detection:** Phishing
- **Credential-Stealing Detection:** Password field detected
- **Final Verdict:** High-Risk Website

These examples demonstrate the system's capability to accurately classify websites and provide actionable insights for users and cybersecurity professionals. By analyzing the structural and behavioral features of websites, *PhishGator* offers a robust and scalable solution for detecting phishing and credential-theft behaviors.

4.4 Detailed Insights from Results

The high accuracy achieved by GBC highlights the importance of selecting advanced machine learning models for phishing detection, as supported by current cybersecurity frameworks [6]. Additionally, the ability to analyze both URL and HTML-based features enhances the system's robustness against sophisticated phishing attacks [2]. Features such as the use of HTTPS, domain registration length, and the presence of suspicious form handlers proved to be the most influential in the detection process.

Furthermore, the system's ability to combine phishing detection with credential-stealing analysis ensures comprehensive protection against cyber threats [5]. This dual-layered approach not only detects deceptive websites but also identifies malicious behaviors such as keylogging and unsafe password collection practices.

The system's performance metrics suggest its suitability for integration into real-world platforms, such as browser extensions and corporate security frameworks [1]. Its scalability and high accuracy make it a viable solution for organizations seeking to enhance their cybersecurity posture.

4.5 Real-World Applications

The results demonstrate the practical applications of the *PhishGator* system in real-world scenarios, following established qualitative research methodologies [7]. For instance, the system can be deployed as:

- A browser extension to provide real-time warnings to users when visiting potentially malicious websites.
- A cybersecurity tool for enterprises to analyze and classify URLs and HTML content in email communications and web applications.
- A training tool for raising awareness among users about phishing and credential-stealing behaviors.

The ability of the system to adapt to new threats and evolving phishing tactics further underscores its value as a comprehensive cybersecurity solution [4].

5 Discussion

5.1 Key Insights

The development and implementation of the *PhishGator* system yielded significant insights into the domains of phishing detection and credential-theft identification [2]. By leveraging a hybrid approach that combines URL-based and HTML-based features, the system effectively addresses the complexity of detecting phishing websites and malicious behaviors.

The results highlighted several important aspects of phishing detection: **1. Importance of Feature Selection:** Features such as domain registration length (`DomainRegLen`) and the presence of HTTPS played a critical role in distinguishing phishing websites from legitimate ones [6]. These features demonstrated strong predictive power, underscoring their relevance in phishing detection models.

2. Integration of Behavioral Analysis: The incorporation of behavioral indicators, such as keylogging scripts and domain mismatches in form submissions, significantly enhanced the detection capabilities of the system [3]. By examining the actions performed on a webpage, *PhishGator* goes beyond static analysis and detects dynamic threats effectively.

3. Gradient Boosting Classifier (GBC): The GBC's performance proved its ability to handle non-linear relationships among features and to manage imbalanced datasets efficiently. Its high accuracy and F1-score showcased its suitability for phishing detection tasks, especially in scenarios involving complex interactions among features.

textbf4. Comprehensive Protection: By integrating phishing detection with credential-stealing detection, the system addresses a broader spectrum of threats [5]. This dual-layered approach ensures a holistic cybersecurity solution capable of identifying deceptive websites and unsafe behaviors simultaneously.

5. Practical Applicability: The system's scalability and accuracy make it ideal for real-world applications such as browser extensions, corporate security tools, and training modules [1]. Its ability to adapt to evolving phishing tactics further enhances its utility in diverse environments.

These insights emphasize the value of adopting a multifaceted approach to phishing detection, leveraging both structural and behavioral analysis to improve accuracy and robustness [2].

5.2 Challenges and Limitations

Despite its effectiveness, the *PhishGator* system faced several challenges and limitations that highlight areas for future improvement, as identified through rigorous qualitative analysis [7]:

1. WHOIS Data Inconsistencies: The reliance on domain registration length (`DomainRegLen`) as a key feature was limited by the availability and accuracy of WHOIS data. Missing or incomplete registration records reduced the effectiveness of this feature in certain cases.

2. Dynamic Content Challenges: Phishing websites often employ JavaScript-heavy or dynamically generated content, which poses challenges for static analysis [4]. While the system performed well on static features, its capabilities could be further enhanced with tools for analyzing dynamic webpage behaviors in real-time.

3. Imbalanced Dataset: Although the Gradient Boosting Classifier demonstrated robustness against imbalanced datasets, future iterations of the system could benefit from exploring advanced techniques for data augmentation or synthetic data generation to address this issue more comprehensively.

4. Limited Training Scope: The datasets used for model training were limited in scope, focusing primarily on known phishing websites [3]. Expanding the training dataset to include more diverse and recent examples, including AI-generated phishing pages, would improve the system's ability to detect emerging threats.

5. Computational Overhead: While GBC achieved high accuracy, it comes with computational overhead that may limit its deployment in resource-constrained environments. Opti-

mizing the model for faster inference without compromising accuracy is an area for future development.

6. Integration Challenges: While the system is suitable for standalone applications, its integration into existing cybersecurity frameworks [6] and workflows may require additional development effort, especially in terms of interoperability and user interface design.

7. False Positives and Negatives: Although the system minimized false positives and negatives, some edge cases remained, particularly involving legitimate websites with unusual configurations or phishing websites mimicking legitimate behaviors. Fine-tuning the system to address such cases is necessary to ensure consistent reliability.

These challenges underscore the need for continuous improvement and adaptation to evolving cybersecurity threats [5]. Future work should focus on addressing these limitations to enhance the system's effectiveness and usability.

6 Conclusion

6.1 Summary

The *PhishGator* system represents a pivotal advancement in the field of phishing detection and credential-theft prevention [6]. Phishing attacks continue to be one of the most prominent cybersecurity threats, exploiting user trust to compromise sensitive data. Addressing this issue required a multifaceted approach that goes beyond traditional detection mechanisms [2]. Through this study, we have demonstrated the effectiveness of integrating structural and behavioral analysis with advanced machine learning techniques to provide a robust and comprehensive solution.

The system's foundation lies in its ability to accurately classify websites as legitimate or phishing using features derived from both URL and HTML content [3]. The Gradient Boosting Classifier (GBC), with a test accuracy of 97%, showcased superior performance in handling imbalanced datasets and identifying complex relationships between features. This level of accuracy is critical for real-world applications, where even minor classification errors can lead to significant consequences for users and organizations.

Moreover, *PhishGator* integrates phishing detection with credential-theft behavior analysis, providing a dual-layered defense mechanism [5]. This integration ensures that even legitimate-looking websites with unsafe practices, such as insecure password forms or keylogging scripts, are flagged as high-risk. The ability to categorize websites into nuanced classifications—safe, phishing, credential-stealing, or both phishing and credential-stealing—enhances the system's applicability across diverse scenarios.

In addition to its technical achievements, the study emphasizes the importance of user-centric design in cybersecurity solutions [7]. By leveraging insights into common phishing behaviors, such as domain mismatches and non-secure form submissions, the system also serves as an educational tool, empowering users to make informed decisions while interacting with websites.

6.2 Implications

The implications of the *PhishGator* system extend across multiple dimensions, from practical applications and research contributions to policy-making and user education [4].

1. Real-World Applications: The system's ability to achieve high accuracy in phishing detection positions it as a valuable tool for deployment in various real-world settings [1]. It can be integrated into web browsers as an extension to provide real-time alerts to users about potentially unsafe websites. Similarly, it can be incorporated into corporate security infrastructures to monitor network traffic, block malicious websites, and safeguard employee credentials from phishing attempts.

2. Enhancing Security Frameworks: Organizations can use *PhishGator* as part of their cybersecurity frameworks to address the growing complexity of phishing attacks [5]. By identifying both structural and behavioral indicators of malicious activity, the system complements existing tools such as firewalls, intrusion detection systems, and endpoint protection software, creating a layered security approach that reduces vulnerabilities.

3. Contributions to Research: The study's methodology and findings make significant contributions to the field of cybersecurity research [2]. By demonstrating the effectiveness of combining URL-based and HTML-based feature analysis with machine learning, this study paves the way for further exploration of hybrid detection techniques. Future researchers can build upon this work to develop more advanced models that address emerging threats, such as AI-generated phishing websites.

4. Educating Users: A key takeaway from the *PhishGator* system is its potential to educate users about phishing threats [4]. By analyzing patterns of phishing behavior, such as excessively long URLs, domain mismatches, and insecure forms, the system provides actionable insights that can be used to develop user awareness programs. These programs can empower individuals to recognize and avoid phishing attempts, thereby reducing their susceptibility to cyberattacks.

5. Supporting Policy and Regulation: The findings from this study can inform policymakers and regulatory bodies in developing standards and guidelines to combat phishing [6]. For instance, the identification of insecure practices, such as non-HTTPS forms and keylogging scripts, underscores the need for stricter web standards. Governments and organizations can leverage these insights to create policies that mandate secure web practices and penalize non-compliance.

6. Addressing Emerging Threats: Phishing techniques are evolving rapidly, with attackers employing sophisticated tactics such as AI-generated phishing websites and dynamic content obfuscation [7]. The *PhishGator* system's hybrid ap-

proach provides a foundation for addressing these challenges. By continuously adapting its feature extraction and detection mechanisms, the system can remain effective against emerging threats, ensuring long-term relevance in the cybersecurity landscape.

7. Ethical Considerations: As phishing detection systems become more integrated into user-facing applications, ensuring fairness and reliability becomes paramount [3]. *PhishGator* achieves this by minimizing false positives and negatives, thereby maintaining user trust. This ethical approach is particularly important in corporate settings, where overly sensitive detection mechanisms can disrupt workflows and hinder productivity.

8. Integration into Diverse Environments: The system's scalability and adaptability make it suitable for deployment in a variety of environments [1]. From individual users and small businesses to large enterprises and government agencies, *PhishGator* can be customized to meet the unique needs of different stakeholders. Its modular design allows for seamless integration into existing cybersecurity architectures, enhancing their overall effectiveness.

9. Future Enhancements: While the current implementation of *PhishGator* has proven effective, there is significant scope for future enhancements [2]. For instance, incorporating real-time detection capabilities and reinforcement learning algorithms can improve the system's adaptability to new phishing techniques. Additionally, expanding the dataset to include a more diverse range of phishing examples, such as those targeting specific industries or user demographics, can further refine the system's accuracy and applicability.

6.3 Final Remarks

The *PhishGator* system represents a significant step forward in addressing the global challenge of phishing and credential theft. By integrating advanced machine learning techniques with comprehensive feature analysis, the system provides a robust and scalable solution that balances technical sophistication with user-centric design. As phishing attacks continue to evolve, tools like *PhishGator* will play a crucial role in safeguarding users, organizations, and critical infrastructures from cyber threats. The findings of this study highlight the importance of continued innovation and collaboration in the fight against phishing, paving the way for a safer and more secure digital ecosystem.

7 Future Work

The PhishGator system has demonstrated significant potential in detecting phishing and credential-stealing behaviors. However, to maintain its relevance and effectiveness, several critical enhancements are envisioned for future iterations [6]:

7.1 Real-Time Detection Integration

Currently, the system operates offline, analyzing pre-collected datasets. The next logical step is to integrate real-time detection capabilities [2], allowing the system to monitor and flag suspicious websites or behaviors dynamically as users browse. This could involve the development of browser extensions or integration with corporate security systems to provide immediate alerts, following established cybersecurity frameworks [5].

7.2 Adapting to Evolving Threats

Phishing tactics evolve rapidly, with attackers employing more sophisticated techniques such as AI-generated websites and dynamic content [4]. To counter this, the system must adopt adaptive learning techniques, such as continuous model retraining or reinforcement learning, to detect novel phishing methods without relying solely on historical data.

7.3 Dynamic Content and Behavior Analysis

Future iterations should move beyond static analysis of URLs and HTML templates to incorporate dynamic behaviors such as JavaScript activity, form submission flows, and AJAX requests [3]. This would enable the system to identify more advanced phishing attacks that rely on real-time obfuscation or user interaction.

7.4 Scalability and Deployment in Diverse Environments

The system's scalability must be rigorously tested in real-world environments, such as enterprise networks, financial institutions, and educational organizations [7]. Tailoring the system to address environment-specific needs, such as high traffic volumes or compliance requirements, will ensure widespread applicability.

7.5 Enhancing Usability and User Awareness

For non-technical users, usability remains a critical factor [1]. Simplified interfaces and actionable alerts can make the system more accessible. Additionally, integrating PhishGator into educational campaigns or cybersecurity training programs can help raise awareness about phishing threats and safe browsing practices.

These focused improvements aim to make PhishGator a versatile and future-ready tool, capable of combating emerging phishing threats and protecting users in diverse scenarios.

8 Study Protocol (Appendix)

8.1 Dataset Details

The datasets used in this study were integral to both training and testing the phishing detection model [4]. The primary dataset, `phishing.csv`, contained 32 features that encapsulated various characteristics of URLs and associated website attributes. These features included indicators such as the presence of an IP address in the URL (`UsingIP`), unusual URL lengths (`LongURL`), the use of obfuscation symbols like `@` (`Symbol@`), and whether the domain included a hyphen (`PrefixSuffix`).

Other critical features captured HTTPS presence (`HTTPS`), iframe redirection (`IFrameRedirection`), and additional aspects known to be indicative of phishing behaviors [2]. The target variable categorized websites as either phishing (-1) or legitimate (1).

A separate dataset, `test_urls.csv`, was used to evaluate the model's performance [3]. It consisted of labeled URLs that helped ensure the model's robustness on unseen data. These datasets were preprocessed to handle missing values, encode categorical features, and normalize the range of numerical attributes, ensuring compatibility with the machine learning algorithms.

8.2 Survey Texts or Templates Used

To validate the effectiveness of the model's detection capabilities and its user-friendliness, a survey was conducted following established qualitative research methodologies [7]. Participants were presented with a series of screenshots from legitimate and phishing websites. They were asked to evaluate the likelihood of each being phishing based solely on visual cues. Additionally, feedback on the system's recommendations, such as its classifications and suggested actions, was collected.

The survey text included clear instructions: Participants were informed about the purpose of the study and provided with a brief explanation of phishing attacks and credential-stealing techniques [6]. They were then asked to rate the clarity and usefulness of the system's outputs, such as its alerts and feature importance metrics.

8.3 Experimental Setup

The experimental framework was designed to ensure repeatability and reliability, following established cybersecurity research practices [5]. The phishing detection model was implemented using Python, leveraging libraries like `pandas` for data preprocessing, `scikit-learn` for model training and evaluation, and `matplotlib` for visualization.

The Gradient Boosting Classifier (GBC) was chosen for its robustness in handling complex classification tasks [2]. The training dataset was split into 80% for training and 20% for validation to ensure that the model was evaluated on unseen data. Hyperparameters, including the depth of decision trees and the learning rate, were optimized using grid search techniques.

The detection pipeline was tested on a virtual machine running Ubuntu 20.04 with 16 GB of RAM and a quad-core processor. This setup allowed the evaluation of the model's efficiency and scalability [4]. Experimental results demonstrated the model's ability to identify phishing websites with an accuracy of 97%, validating its practical applicability in real-world scenarios.

Additional measures included testing the model's performance on diverse datasets to assess its generalization capabilities and its ability to adapt to emerging phishing tactics [1].

References

[1] NIST Cybersecurity Framework Adoption Linked to Higher Security Confidence According to New Research from Tenable Network Security, March 2016.

[2] Riza Azmi, William Tibben, and Khin Than Win. Review of cybersecurity frameworks: context and shared concepts. *Journal of Cyber Policy*, 3(2):258–283, May 2018. Publisher: Routledge _eprint: https://doi.org/10.1080/23738871.2018.1520271.

[3] Glenn A. Bowen. Document Analysis as a Qualitative Research Method. *Qualitative Research Journal*, 9(2):27–40, August 2009.

[4] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, Inc., 2019.

[5] National Institute of Standards and Technology. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. Technical Report NIST CSWP 04162018, National Institute of Standards and Technology, Gaithersburg, MD, April 2018.

[6] National Institute of Standards and Technology. The NIST Cybersecurity Framework 2.0 (Draft). Technical Report NIST CSWP 29, U.S. Department of Commerce, August 2023.

[7] Sarah J. Tracy. Qualitative Quality: Eight "Big-Tent" Criteria for Excellent Qualitative Research. *Qualitative Inquiry*, 16(10):837–851, December 2010. Publisher: SAGE Publications Inc.