International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

PHISHCATCHER: CLIENT-SIDE DEFENSE AGAINST WEB SPOOFING ATTACKS USING MACHINE LEARNING

D.Shine Rajesh[1], K.Yashaswi[2], L.Sriya Varma[3], N.Bindu Madhavi Devi[4], T.Aparna[5]
[1] Assistant Professor, Department of IT,Malla Reddy Engineering College For Women(Autonomous Institution), Maisammaguda,Dhulapally,Secunderabad,Telangana-500100
[2345]UG Scholar, Department of IOT,Malla Reddy Engineering College for Women, (Autonomous Institution), Maisammaguda,Dhulapally,Secunderabad,Telangana-500100
Email: shinerajesh@gmail.com

ABSTRACT

Cybersecurity faces a major challenge in maintaining the confidentiality and integrity of personal user information such as passwords and PIN codes. Every day, billions of users are exposed to fake login pages that request sensitive information. There are many ways to trick users into visiting websites, such as phishing emails, bait-and-switch ads, clickjacking, malware, SQL injection, session hijacking, man-in-the-middle, denial of service, and cross-site scripting attacks. Web spoofing, or phishing, is an electronic trick in which an attacker creates a malicious copy of a legitimate webpage and requests personal user information such as passwords. To combat such attacks, researchers have proposed several security strategies, but they suffer from latency and accuracy issues. To overcome such problems, we propose and develop a client-side defense mechanism based on machine learning techniques to detect fake websites and protect users from phishing attacks. As a proof of concept, a Google Chrome extension called PhishCatcher is developed that implements a machine learning algorithm to classify URLs as suspicious or trustworthy. The algorithm takes four different types of web features as input and uses a random forest classifier to determine whether a login web page is fake or not. To evaluate the accuracy and precision of the extension, several experiments were conducted on real web applications. The experimental results showed an astonishing accuracy of 98.5% and a precision of 98.5% from experiments on 400 classified phishing URLs and 400 legitimate URLs. To measure the latency of the tool, experiments with more than 40 phishing URLs were also conducted. The average response time recorded for PhishCatcher was only 62.5 milliseconds.

Keywords-Phishing Detection, Web Spoofing, Client-Side Defense, Cybersecurity, Phishing Prevention

I. INTRODUCTION

With the rapid advancement in digital technologies, cybercrime is fast becoming common, more especially in the phishing and web spoofing attacks. Phishing is a type of attack where attackers pretend to be actual legitimate entities for the purpose of accessing one's personal information that can be highly sensitive. Attacks such as phishing usually start through spam emails, which hold links to fraudulent websites masquerading as the correct website. Once the credentials of users have been entered by them, attackers can hijack them for the malicious purpose of identity theft, financial loss, or unauthorized access to confidential information.

One of the significant phishing attacks was in October 2022, when officials of National Institute for Research in Digital Science and Technology (Inria) in France received an official communication from the attackers in the phishing email. The email contained a malicious link that took the user to a phishing login page, which was almost identical to the real Inria login page. The users were unaware of the scam and entered their login credentials, which were then captured by the attackers. This scenario depicts the serious threat web spoofing poses, especially in environments where sensitive data is involved.
With phishing attacks on the rise, researchers have been developing novel solutions for detecting and mitigating this kind of attack. A key such technique includes applying machine learning-based algorithms for classifying malicious web pages using their inherent properties. One such client-side approach is PhishCatcher-a Google Chrome extension designed to thwart web spoofing attacks. Utilizing Random Forest machine learning algorithms, PhishCatcher can distinguish between a legitimate login page and a phishing attempt. This tool analyzes key web features and classifies suspicious pages, which offers the user a proactive defense mechanism.

As web spoofing continues to evolve, with tactics such as QR code phishing and spear-phishing, protection of users from these attacks is becoming increasingly important. Tools such as PhishCatcher show the potential of machine learning in creating robust, client-side defenses that do not require changes to the server. This is particularly valuable in real-time detection of phishing websites, preventing users from falling victim to online scams. Such solutions shall require implementation because, along with the cyber threats that keep advancing every day, a digital world requires trust and security in online transactions and services.

II. RELATED WORK

1. SpoofCatch: A Client-Side Protection Tool Against Phishing Attacks (W. Khan et al., 2021)

Khan et al. (2021) introduce Spoof catch , a client-side tool designed to mitigate phishing attacks. This tool employs sophisticated mechanisms to identify deceptive websites, ensuring that users are protected while browsing. Their research highlights the effectiveness of integrating client-side security measures to prevent phishing, which is critical as traditional server-side methods often fail to catch all phishing attempts.

2. Two-Factor Authentication: Too Little, Too Late (B. Schneier, 2005)

In this seminal paper, Schneier (2005) discusses the limitations of two-factor authentication (2FA) as a protective measure against phishing. While 2FA is seen as a critical enhancement in security, Schneier argues that it might be insufficient to fully prevent phishing attacks, emphasizing the need for additional layers of defense beyond just authentication.

3. A Framework for Detection and Measurement of Phishing Attacks (S. Garera et al., 2007)

Garera et al. (2007) propose a framework for detecting phishing attacks. This approach utilizes statistical techniques to measure phishing incidents, presenting a comprehensive methodology for understanding and addressing phishing's various facets. Their work is foundational in creating detection systems that go beyond basic heuristics, offering a detailed analysis of phishing characteristics.

4. Effective Protection Against Phishing and Web Spoofing (R. Oppliger & S. Gajek, 2005)

This study explores various strategies for protecting users against web spoofing and phishing. Oppliger and Gajek (2005) focus on improving security measures on the web, specifically through browser enhancements and URL validation techniques. They emphasize the importance of ensuring that users can recognize legitimate sites through various security indicators, such as digital certificates.

5. Defending Against Injection Attacks Through Context-Sensitive String Evaluation (T. Pietraszek & C. V. Berghe, 2005)

Pietraszek and Berghe (2005) address a different aspect of web security: defending against injection attacks. While not directly related to phishing, their work is significant as many phishing attacks exploit web vulnerabilities, including SQL injections. Their method of context-sensitive string evaluation provides insights into improving web application security, which can be applied in the broader context of preventing phishing.

III.IMPLEMENTATION

1.System Design and Architecture

First in implementation, design system architecture that contains structure in phishing detection engine consisting of modules for analysis on URL, web page content, and visual similarity. There is also an ongoing development of the client-side protection tool's user interface, such as browser extensions and desktop applications, and a warning system to notify the user that a phishing site has been detected.

2.Collection of data and preprocessing

Data collection to identify phishing sites. It is a process in which records are gathered from identified phishing sites, like those on PhishTank or any other phishing database. The data gathered is preprocessed to extract all domain characteristics, URL patterns, and other relevant content elements. This information is also flagged with one of the two states- phishing or legitimate-while for machine learning models it is cleaned and normalized in order to further train.

3.Phishing sit detection algorithms implementation.

At this stage, the actual algorithms to detect phishing sites are designed. These include:

URL-based detection: It includes extracting domain names, looking for suspicious patterns, and correlating URLs with known phishing databases.

Content-based detection: Analysis of the content on a web page includes HTML, forms, links to detect suspicious characteristics such as missing registration forms or missing HTTPS.

Visual similarity detection: Use techniques of image processing or deep learning models, comparing the layout of a page with known legitimate web pages.

4.Training the machine learning model

The model is trained on the data which has been labeled using the machine learning techniques of either phishing or legitimate websites. URL length, domain name patterns, and content characteristics are fed into the model so that it may classify new websites as either phishing or legitimate. Its performance is evaluated with accuracy, precision, and recall using a dataset like PhishTank.

5.Creating a real-time guard

Once the detection engine is in place, create a client-side protection tool. It could be in the form of an Internet browser extension or desktop application running in the background while the user is on the Internet. It shall work in tandem with the phishing detection engine by assessing visited websites and send notifications whenever it finds out sites that are phishing attacks.

6. Integrating external APIs

To improve detection, the system can integrate external phishing detection APIs, such as the PhishTank API or Google Safe Browsing API. These APIs provide a database of known phishing sites. Incorporating them allows the tool to quickly check visited URLs against these lists and improve recognition accuracy.

7. User Interface Development

The user interface is developed to make it easy for the user to operate the phishing protection tool. This may include: Design a basic warning system to warn the user that they are browsing an undesirable site.Present the user with a dashboard showing the status of phishing protection and the last detection. Ensure that the user interface does not interfere with the normal user experience.

8. Testing and Evaluation

Testing should be comprehensive enough before deployment.This would include:

Unit testing: Confirm that all components such as URL parsing, content detection, and machine learning models are working correctly.

Integration testing: Confirm that all components of the phishing detection system are working as a unit. **Performance testing:** Confirm that the whole system responds very fast and does not bog down the user's general browsing experience.

Evaluation metrics: The working of the recognition engine relies on metrics calculated through the use of accuracy, precision, recall, and F1 score. The tool must choose phishing sites which have few or no false positives or negatives.

9. Deployment

Once the system is thoroughly tested and ready, then it is deployed. It is submitted to a store for browser extensions. Stores include the Chrome Web Store or Firefox add-on. It can be downloaded from a website or app store for desktop applications.

IV.ALGORITHMS USED

1. URL Analysis Algorithms

URL analysis algorithms focus on examining the structure and components of URLs to detect phishing attempts. For example, domain name analysis checks if a URL contains misspelled or suspicious domain names that resemble legitimate sites. The entropy-based algorithm calculates the randomness of a URL to detect patterns typically found in phishing URLs. The entropy formula is used here:

$$H(x) = -\sum p(x) \log p(x)$$

where p(x) is the probability of each character in the URL. Another common method involves **blacklist checking**, where URLs are cross-referenced with known phishing site databases like Google Safe Browsing

2. Content-Based Detection Algorithms

Content-based detection algorithms analyze the content of a webpage to identify suspicious patterns. Heuristic rule-based algorithms use predefined rules to detect elements such as fake login forms or missing HTTPS encryption. Content similarity algorithms compare textual content between a suspicious website and legitimate sites. One popular approach is using Cosine Similarity, calculated as:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

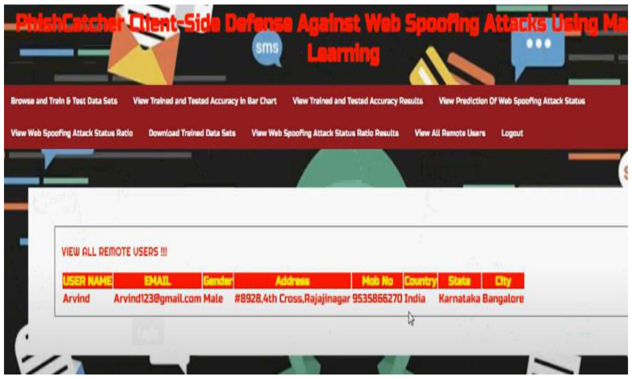
where Aand B are term frequency vectors representing the content of two websites.

MACHINE LEARNING ALGORITHMS

- Logistic Regression is one of the simplest machine learning models used for binary classification. It predicts the probability of a site being phishing or legitimate by calculating the weighted sum of features, followed by a logistic function to output a value between 0 and 1.
- Decision Trees work by recursively splitting data into subsets based on the most significant feature at each node, using a set of decision rules. The tree classifies sites by following these rules. Features such as the length of the URL or the presence of suspicious keywords are typically used to make these splits.
- Random Forest is an ensemble learning technique that builds multiple decision trees. It combines the results of these trees by taking a majority vote (in classification tasks) to improve accuracy and reduce the risk of overfitting that is often seen with individual decision trees.
- Support Vector Machines (SVM) create an optimal hyperplane in a multi-dimensional space that best separates the data into two classes (phishing and legitimate). SVMs work well when there is a clear margin of separation between classes, using different kernel functions to handle non-linear data.
- Naive Bayes is a probabilistic classifier based on Bayes' theorem, which calculates the probability of a website being phishing based on the presence of specific features. It assumes that all features are independent, making it a simple yet effective model for phishing detection.

V.RESULTS

Fig:1:User Login


PhishCatcher Client-Side Defense Against Web Spoofing Attacks Using Machine Learning

Browse and Train & Test Data Sets | View Trained and Tested Accuracy in Bar Chart | View Trained and Tested Accuracy Results | View Prediction Of Web Spoofing Attack Status | View Web Spoofing Attack Status Ratio | Download Trained Data Sets | View Web Spoofing Attack Status Ratio Results | View All Remote Users | Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Arvind	Arvind123@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore

Fig:2:Remote Users

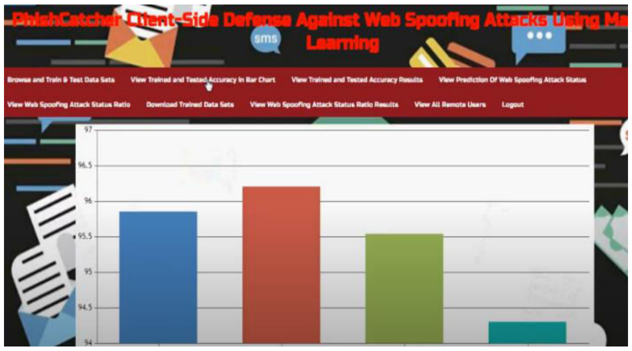


Fig:3:Tested Accuracy Results

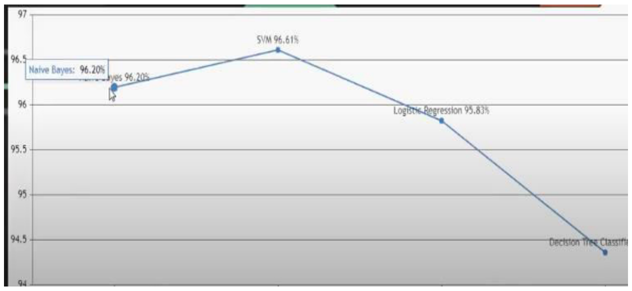


Fig:4:Accuracy

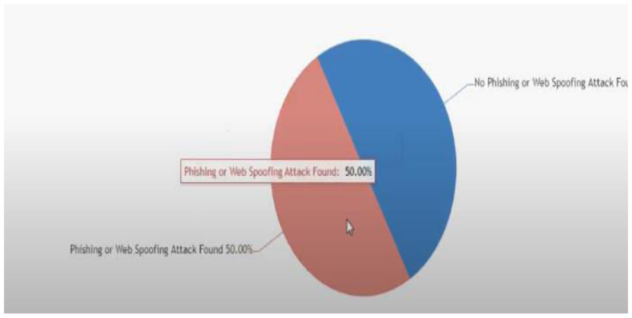
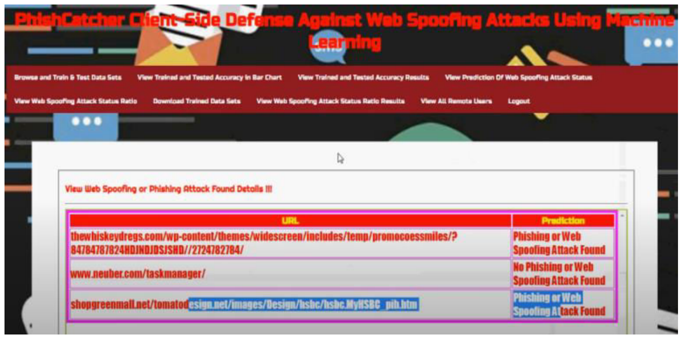


Fig:5:Phishing /Web Attack


PhishCatcher Client-Side Defense Against Web Spoofing Attacks Using Machine Learning

Browse and Train & Test Data Sets | View Trained and Tested Accuracy in Bar Chart | View Trained and Tested Accuracy Results | View Prediction Of Web Spoofing Attack Status | View Web Spoofing Attack Status Ratio | Download Trained Data Sets | View Web Spoofing Attack Status Ratio Results | View All Remote Users | Logout

View Web Spoofing or Phishing Attack Found Details !!!

URL	Prediction
thewhiskeydregs.com/wp-content/themes/widescreen/includes/temp/promocoessmiles/?84784787824HDJNDJDSJSHD//2724782784/	Phishing or Web Spoofing Attack Found
www.neuber.com/taskmanager/	No Phishing or Web Spoofing Attack Found
shopgreenmall.net/tomatodesign.net/images/Design/hsbc/hsbc.MyHSBC_pib.htm	Phishing or Web Spoofing Attack Found

Fig:6:Phishing /Web Attack details

VI.CONCLUSION

Users have become dependent on online applications because they provide high-quality services in many areas, such as: B. online banking, e-commerce, social networking, digital libraries, online medical services, virtual education, digital marketing, and multiplayer gaming applications. Typically, users go through an authentication process to create online accounts and access private web content. In the face of sophisticated web spoofing attacks, users' security and privacy are at risk. Several research and commercial tools have been developed to combat web spoofing attacks, but most of them have some shortcomings. We have developed a streamlined, easy-to-use browser plug-in called Phish Catcher that leverages supervised machine learning to intelligently detect phishing attacks.

Unlike traditional approaches, our scheme provides the ability to perform classification in the browser itself. It addresses gaps in existing web applications by fixing latency issues and improving the efficiency of the tool. The interface of our plugin is designed to be simple for users to understand it better. When a user enters a phishing URL, a phishing warning is displayed on the screen and the phishing features corresponding to that URL are highlighted in a drop-down menu. The feature set contains 30 features categorized into four groups, and each group is called a decision tree. The random forest classifier uses the aggregated results of the decision tree to identify fake and genuine login websites. The test and evaluation dataset contains 400 malicious URLs and 400 genuine URLs. The test and evaluation criteria are based on a confusion matrix that lists true positives, true negatives, false positives, and false negatives. Our plugin showed impressive classification results with a precision and recall of 98.5% and a precision of 98.5%, respectively. In addition, the average latency of the plugin measured by running over 40 phishing URLs was only 62.5 ms. Although the feature set contains 30 features, we recommend adding more automation features to improve the overall performance. Several other discriminative classifiers, such as SVM, can also be implemented to predict fake or genuine URLs by training on a larger dataset. To improve performance analysis, evaluation criteria can also be further developed using various tools.

REFERENCES

[1] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, ‘‘SpoofCatch: A client-side protection tool against phishing attacks,’’Mar. 2021.

[2] B. Schneier, ‘‘Two-factor authentication: Too little, too late,’’ Apr. 2005.

[3] S. Garera, N. Provos, M. Chew, and A. D. Rubin, ‘‘A framework for detection and measurement of phishing attacks,’’ Nov. 2007.

[4] R. Oppliger and S. Gajek, ‘‘Effective protection against phishing and web spoofing,’’ 2005.

[5] T. Pietraszek and C. V. Berghe, ‘‘Defending against injection attacks through context-sensitive string evaluation,’’ 2005.

[6] M. Johns, B. Braun, M. Schrank, and J. Posegga, ‘‘Reliable protection against session fixation attacks,’’ 2011.

[7] M. Bugliesi, S. Calzavara, R. Focardi, andW. Khan, ‘‘Automatic and robust client-side protection for cookie-based sessions,’’ 2014.

[8] A. Herzberg and A. Gbara, ‘‘Protecting (even naıve) web users from spoofing and phishing attacks,’’ 2004.

[9] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, ‘‘Client-side defense against web-based identity theft,’’2004.

[10] B. Hämmerli and R. Sommer, Detection of Intrusions and Malware, and Vulnerability Assessment July 12-13, 2007

[11] C. Yue and H. Wang, ‘‘BogusBiter: A transparent protection against phishing attacks,’’ May 2010.

[12] W. Chu, B. B. Zhu, F. Xue, X. Guan, and Z. Cai, ‘‘Protect sensitive sites from phishing attacks using features extractable from inaccessible
phishing URLs,’’ Jun. 2013,

[13] Y. Zhang, J. I. Hong, and L. F. Cranor, ‘‘Cantina: A content-based approach to detecting phishing web sites,’’ May 2007.

[14] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, ‘‘An evaluation of machine learning-based methods for detection of phishing sites,’’ 2008.

[15] E. Medvet, E. Kirda, and C. Kruegel, ‘‘Visual-similarity-based phishing detection,’’ Sep. 2008.
[16] W. Zhang, H. Lu, B. Xu, and H. Yang, ‘‘Web phishing detection based on page spatial layout similarity,’’ 2013.
[17] J. Ni, Y. Cai, G. Tang, and Y. Xie, ‘‘Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics,’’ Oct. 2021.
[18] W. Liu, X. Deng, G. Huang, and A. Y. Fu, ‘‘An antiphishing strategy based on visual similarity assessment,’’Mar. 2006.
[19] A. Rusu and V. Govindaraju, ‘‘Visual CAPTCHA with handwritten image analysis,’’ 2005.
[20] P. Yang, G. Zhao, and P. Zeng, ‘‘Phishing website detection based on multidimensional features driven by deep learning,’’ 2019.
[21] P. Sornsuwit and S. Jaiyen, ‘‘A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting,’’ 2019.
[22] S. Kaur and S. Sharma, ‘‘Detection of phishing websites using the hybrid approach,’’ 2015.