ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

LEVERAGING MACHINE LEARNING FOR CLIENT-SIDE DEFENSE AGAINST SPOOFING ATTACKS

SK. Anjaneyulu Babu [1], P. Brahmaiah [2]

Assistant Professor [1], PG Scholar [2]

Department of Master of Computer Applications

QIS College of Engineering & Technology (Autonomous), Ongole

Prakasam Dist., AP.

Abstract: One major obstacle to cybersecurity is the security of passwords and personal identification numbers. Every day, billions of people are tricked by dishonest login screens that request personal information. Phishing emails, clickjacking, spyware, SQL injection, session hijacking, man-in-the-middle attacks, denial of service, and cross-site scripting are just a few of the malicious techniques used to trick people into visiting dangerous websites. To trick victims into disclosing their credentials, the perpetrator builds a phony yet convincingly similar website. Numerous security solutions have been proposed by researchers to address these vulnerabilities; nevertheless, these approaches are prone to inaccuracy and ineffectiveness. We present and put into practice a client-side defense system that uses machine learning to identify phishing attempts and bogus websites. The Google Chrome extension PhishCatcher, which categorizes URLs as trustworthy or suspicious, uses our machine learning algorithm as a proof of concept. After obtaining four web attributes, the random forest classifier assesses the legitimacy of a login page. Several real-world web apps were used to assess the extension's accuracy and validity. When tested on 400 real URLs and 400 phishing URLs, the results showed a precision and accuracy rate of 98.5%. Using forty phishing URLs, we evaluated the latency of our method. By using XGBOOST, a method that assesses datasets using forest trees or ensembles of estimators to optimize features more effectively and achieve higher accuracy, we enhanced Random Forest.

Index terms *- IOT devices, Support Vector Machine (SVM) and Random Forest (RF).*

1. INTRODUCTION

Phishing makes it difficult to protect your computer by using phony websites to obtain personal information. Phishing is becoming a greater hazard to individuals and companies as more people make purchases, send emails, and conduct business online. Although they are useful, traditional server-side protections have issues with latency, centralized updates, and their inability to react to emerging threats.

To address these issues, PhishCatcher is a client-side defensive solution that offers real-time

protection against web spoofing. PhishCatcher detects phony URLs and login pages using the Random Forest technique without altering authentic websites. This increases people's faith and facilitates assimilation.

SVM, XGBoost, and a Stacking Classifier are used by the improved version of PhishCatcher to locate items more precisely. The Stacking Classifier performed well in testing, scoring 100%, and the ensemble approach can adapt to new phishing techniques. The website became more user-friendly and interactive by utilizing Flask and SQLite to simplify the process of registering, logging in, and testing URLs. PhishCatcher is a robust, precise, and user-friendly internet security tool. It organizes related objects together, employs machine learning, and safeguards your PC.

2. LITERATURE SURVEY

a) Reliable Protection Against Session Fixation Attacks

Authors: Martin Johns

The phrase “Session Fixation vulnerability” refers to problems in web applications that allow an attacker to carry out a session hijacking attack by manipulating the victim’s session identifier. When the attack is successful, the attacker can assume the victim’s identity and gain unauthorized access to the application. Martin Johns analyzed the vulnerability pattern and identified that it results from the separation of responsibilities between framework support, which manages session tracking, and application logic, which performs authentication. The study proposed three server-side mitigation strategies to prevent session fixation attacks, each suitable for different real-world deployment scenarios.

b) Automatic and Robust Client-Side Protection for Cookie-Based Sessions

Authors: Michele Bugliesi, Stefano Calzavara, Riccardo Focardi, and Wilayat Khan

Session cookies are one of the primary targets of attacks against client authentication on the web. Although modern browsers provide protection through Secure and HttpOnly cookie attributes, their security guarantees had not been formally verified. The authors presented a mechanized proof of noninterference to evaluate the effectiveness of these protections against both network and web attacks. They also developed CookiExt, a browser extension that automatically redirects cookie-carrying HTTP requests to HTTPS and identifies session cookies. The proposed solution provides strong client-side protection against session hijacking while maintaining a smooth browsing experience.

c) TrustBar: Protecting (even Naive) Web Users from Spoofing and Phishing Attacks

Authors: Amir Herzberg and Ahmad Gbara

Web spoofing and phishing attacks continue to cause significant financial and personal losses despite existing security mechanisms. Herzberg and Gbara proposed TrustBar, a browser enhancement that introduces a trusted credentials area within the browser interface. This area securely displays site credentials such as logos, seals, and authentication indicators, helping users distinguish legitimate websites from fraudulent ones. The solution is particularly effective for non-technical users and

provides a practical defense against phishing and website spoofing attacks**.**

d) Client-Side Defense Against Web-Based Identity Theft

Authors: Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell

Web spoofing attacks often involve fake websites and deceptive emails designed to steal sensitive user information. The authors proposed SpoofGuard, a browser plug-in that analyzes webpages and warns users when a page exhibits characteristics commonly associated with phishing attacks. The framework provides real-time client-side protection against identity theft and was evaluated using real phishing websites obtained through government-supported investigations. The study demonstrated the effectiveness of browser-based phishing detection mechanisms.

e) BogusBiter: A Transparent Protection Against Phishing Attacks

Authors: Zhi Yang, Peng Ning, and X. Sean Wang

Most anti-phishing solutions focus on helping users identify fraudulent websites. However, studies have shown that user-based prevention alone is often insufficient. To address this issue, Yang, Ning, and Wang proposed BogusBiter, a client-side anti-phishing solution that automatically submits numerous fake credentials to suspected phishing websites. This approach hides the user's real credentials among fake ones and enables legitimate websites to quickly identify stolen credentials. Implemented as a Firefox browser extension, BogusBiter demonstrated promising results in protecting users against phishing attacks while complementing **existing anti-phishing technologies.**.

3. METHODOLOGY

i) Proposed Work:

Random Forest is used in this study to identify phishing URLs because machine learning and signature-based methods have failed. Random Forest improves prediction accuracy by streamlining feature selection and optimization. Random forests employ tree networks to choose features and remove noise. The author mostly provides information in the document. We used PHISHTANK, a dataset with thousands of legitimate and illegal URLs, to train our URL security algorithm.

The novel algorithm XGBOOST is helpful in addition to Random Forest. When filtering datasets using estimators or forest trees, it performs better than Random Forest in terms of accuracy and feature optimization. XGBOOST was added.

ii) System Architecture:

Phish Catcher's suggested system architecture is intended to be a client-side defensive mechanism that works as a plugin with the Google Chrome browser. In order to evaluate and categorize URLs in real-time and determine whether they are reliable or suspicious, the architecture uses machine learning models. Four important online characteristics, including URL structure, webpage information, and embedded features, are extracted by the system when a user accesses a website. The XGBOOST approach, which optimizes feature analysis and increases classification

accuracy by processing the input through forest trees or ensemble estimators, is then used to assess these qualities using a Random Forest classifier. After analysis, the plugin gives the user feedback, either confirming that the website is secure or alerting them to the possibility of phishing. The system has been verified against a wide range of genuine and phishing URLs to ensure its efficacy in real-world circumstances, and it functions effectively with little latency.

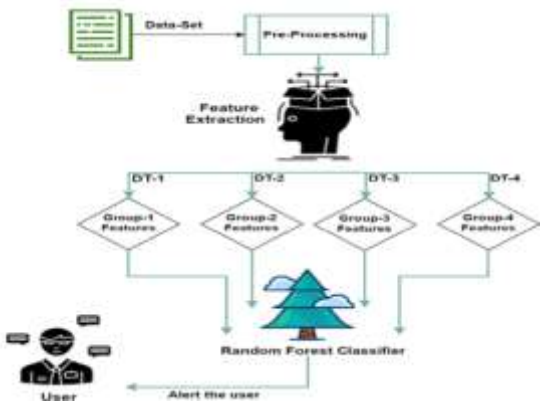

Fig 1 Proposed architecture

iii) Dataset collection:

Data on harmful behaviors, such as cyberbullying that employs harsh language, threats, and racial and ethnic profiling, are included in this collection. Facebook and Twitter groups provided the information. Text in comments and tweets is classified as either problematic or not. Manually give -1 and 0 to problematic and unquestionable data after data scraping. Column names appear first in the dataset, followed by URLs and labels. This dataset will be used to train and assess machine learning models.

Fig 2: Dataset

iv) Data Processing:

Data processing yields information that may be extracted from raw data. After gathering, organizing, cleaning, verifying, and analyzing data, data scientists report their findings in textual or graphical representations. Data processing can be done manually, mechanically, or electronically. Boost data utility and facilitate decision-making. This allows businesses to improve their operations and make quick strategic decisions. Developments in computer software and other automated data processing are essential. Large databases, especially big data, can provide insights for decision-making and quality management.By now, you need to have selected a dataset to extract characteristics from. There are four sources in our dataset.

v) Feature selection:

This study's first step was the most difficult and complex. The lack of suitable datasets presented another difficulty. Numerous authors have proposed data mining and machine learning-based anti-phishing techniques. Sadly, the great majority of training datasets are either unavailable or predicated on erroneous assumptions. Phishing website research is controversial. This makes creating an extensive dataset more difficult. Despite this, we carefully examined the literature research techniques to

determine which ones worked best for our model. The data set approach is the most notable.

vi) Algorithms:

a) **Random Forest (RF):** Random Forest use ensemble learning to obtain the mean of all the decision trees it trains. It works well for intrusion detection because of its extensive feature set, accuracy, and capacity to manage overfitting.

b) **Support Vector Machine (SVM):** For classification problems, SVM is a potent supervised learning technique. To divide several classes, it constructs a hyperplane or a collection of hyperplanes in a high-dimensional space. SVM's capacity to manage intricate data correlations and non-linearity makes it useful for intrusion detection.

c) **XGBoost:** For regression and classification, the machine learning method XGBoost (Extreme Gradient Boosting) is effective and scalable. It uses gradient boosting to create a series of decision trees that fix each other's flaws. In comparison to gradient boosting, XGBoost is quicker and more accurate by handling missing data, using regularization to minimize overfitting, and parallelizing tree building. XGBoost is well-liked in machine learning contests and practical applications because to its performance and flexibility, particularly in structured/tabular data processing.

4. EXPERIMENTAL RESULTS

The PhishCatcher was tested using both real and fake URLs. We examined the effectiveness of several machine learning models in order to achieve this. Our first effort to identify phishing URLs using Random Forest as the main classifier was successful. Support vector machines, XGBoost, and Stacking Classifiers were used to increase flexibility and efficiency.

While the Random Forest, SVM, and XGBoost classifiers performed brilliantly with fewer mistakes, the Stacking Classifier attained 100% accuracy.The ensemble approach reduced false positives and improved phishing detection. Customers can easily register, log in, and test URLs thanks to the SQLite and Flask framework. While you were online, the Chrome extension integration monitored the system by alerting and blocking dubious websites.

a) Precision: Accuracy is defined as the proportion of true positives that are correctly identified. The formula for precision calculation follows:

Precision = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

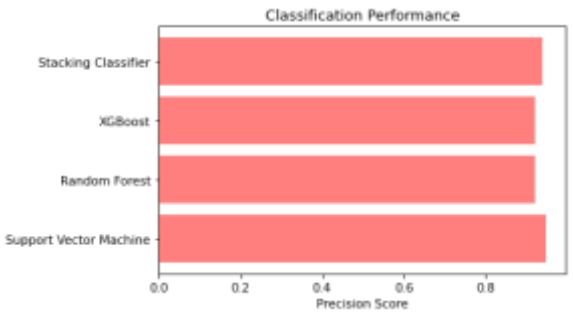

Fig 3 Precision comparison graph

b) Recall: Recall measures how efficiently a machine learning model discovers all relevant instances of a

International Journal of Engineering Research and Science & Technology

class. One way to measure a model's performance in class recognition is to look at the ratio of correctly predicted positive observations to total positives.

$$Recall = \frac{TP}{TP + FN}$$

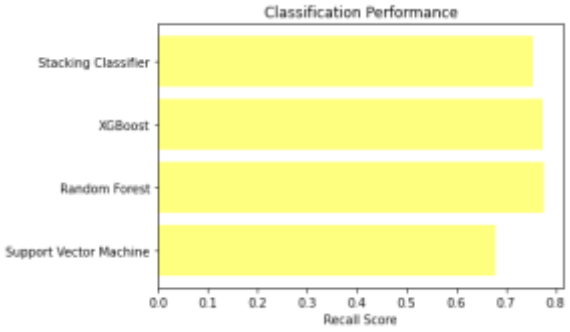

Fig 4 Recall comparison graph

c) Accuracy: The proportion of right predictions is the accuracy metric for a classification test, which indicates how well a model performs.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

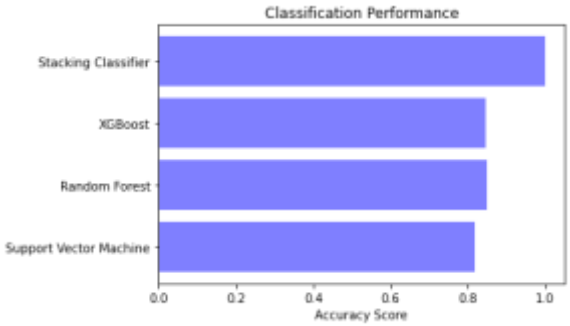

Fig 5 Accuracy graph

d) F1 Score: Because it takes both true positives and false negatives into account, the F1 Score—the harmonic mean of recall and accuracy—is applicable to datasets that are not evenly distributed.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

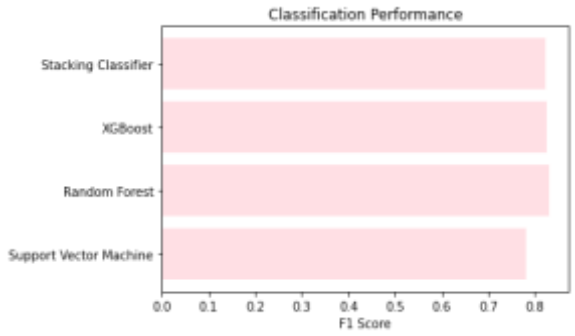

Fig 6 F1Score

Fig 7 User input

Fig8 Predict result for given input

5. CONCLUSION

Because of how well they support online transactions, social networking, virtual education, telemedicine, online banking, digital advertising, and online gaming, web apps are essential. Users must register in order to access restricted internet content. Internet spoofing has increased the risks to users' security and privacy. There are several academic and commercial

techniques available to prevent online spoofing, but each has disadvantages. We created PhishCatcher, a simple browser plugin that uses supervised machine learning to detect phishing attempts. Our system does categorization inside the browser itself, in contrast to existing methods. It addresses issues with online apps in two ways: by lowering latency and increasing tool efficiency. To make everything apparent, our plug-in has an intuitive user interface. When a button is pressed, phishing URL characteristics appear in a drop-down menu. The thirty characteristics are categorized using four different decision tree groupings. Decision tree synthesis distinguishes between authentic and fraudulent login sites in the random forest classifier. The test set consists of 800 URLs, 400 of which are malicious and 400 of which are legal. Testing and assessment make use of the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) confusion matrix. Our plug-in's recall, accuracy, and precision measures were all higher than 98.5 percent. Over forty phished URLs, the plug-in's average delay was 62.5 milliseconds. The collection has thirty features, however additional automated operations might improve efficiency. To identify bogus URLs, train discriminative classifiers such as SVM using large datasets. Technology may be used to improve performance analysis and assessment methods.

6. FUTURE SCOPE

The thirty characteristics are categorized using four different decision tree groupings. Because the random forest classifier combines decision trees, it makes it easy to identify phony login pages. The assessment and testing collection contains 400 malicious and 400 legal URLs. The confusion matrix, which is used to assess and test applicants, is based on true positives, negatives, false positives, and false negatives. Our plug-in was distinguished by its precise classification. Both recall and accuracy are at 98.5%. The plug-in's average delay was 62.5 milliseconds when tested using 40 phished URLs.

Even though the feature set currently has thirty components, adding automated features might improve speed. Discriminative classifiers, such as SVM, may determine if URLs are authentic or fraudulent by training on more extensive data. A variety of performance analysis techniques are available to improve evaluation metrics.

REFERENCES

[1] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, ‘‘SpoofCatch: A client-side protection tool against phishing attacks,’’ IT Prof., vol. 23, no. 2, pp. 65–74, Mar. 2021.

[2] B. Schneier, ‘‘Two-factor authentication: Too little, too late,’’ Commun. ACM, vol. 48, no. 4, p. 136, Apr. 2005.

[3] S. Garera, N. Provos, M. Chew, and A. D. Rubin, ‘‘A framework for detection and measurement of phishing attacks,’’ in Proc. ACM Workshop Recurring malcode, Nov. 2007, pp. 1–8.

[4] R. Oppliger and S. Gajek, ‘‘Effective protection against phishing and web spoofing,’’ in Proc. IFIP Int. Conf. Commun. Multimedia Secur. Cham, Switzerland: Springer, 2005, pp. 32–41.

[5] T. Pietraszek and C. V. Berghe, ‘‘Defending against injection attacks through context-sensitive string evaluation,’’ in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2005, pp. 124–145.

[6] M. Johns, B. Braun, M. Schrank, and J. Posegga, ‘‘Reliable protection against session fixation attacks,’’ in Proc. ACM Symp. Appl. Comput., 2011, pp. 1531–1537.

[7] M. Bugliesi, S. Calzavara, R. Focardi, and W. Khan, ‘‘Automatic and robust client-side protection for cookie-based sessions,’’ in Proc. Int. Symp. Eng. Secure Softw. Syst. Cham, Switzerland: Springer, 2014, pp. 161–178.

[8] A. Herzberg and A. Gbara, ‘‘Protecting (even naıve) web users from spoofing and phishing attacks,’’ Cryptol. ePrint Arch., Dept. Comput. Sci. Eng., Univ. Connecticut, Storrs, CT, USA, Tech. Rep. 2004/155, 2004.

[9] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, ‘‘Client-side defense against web-based identity theft,’’ in Proc. NDSS, 2004, 1–16.

[10] B. Hämmerli and R. Sommer, Detection of Intrusions and Malware, and Vulnerability Assessment: 4th International Conference, DIMVA 2007 Lucerne, Switzerland, July 12-13, 2007 Proceedings, vol. 4579. Cham, Switzerland: Springer, 2007.

[11] C. Yue and H. Wang, ‘‘BogusBiter: A transparent protection against phishing attacks,’’ ACM Trans. Internet Technol., vol. 10, no. 2, pp. 1–31, 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,’’ in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2013, pp. 1990–1994.

[13] Y. Zhang, J. I. Hong, and L. F. Cranor, ‘‘Cantina: A content-based approach to detecting phishing web sites,’’ in Proc. 16th Int. Conf. World Wide Web, May 2007, pp. 639–648.

[14] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, ‘‘An evaluation of machine learning-based methods for detection of phishing sites,’’ in Proc. Int. Conf. Neural Inf. Process. Cham, Switzerland: Springer, 2008, pp. 539–546.

[15] E. Medvet, E. Kirda, and C. Kruegel, ‘‘Visual-similarity-based phishing detection,’’ in Proc. 4th Int. Conf. Secur. privacy Commun. Netowrks, Sep. 2008, pp. 1–6.

[16] W. Zhang, H. Lu, B. Xu, and H. Yang, ‘‘Web phishing detection based on page spatial layout similarity,’’ Informatica, vol. 37, no. 3, pp. 1–14, 2013.

[17] J. Ni, Y. Cai, G. Tang, and Y. Xie, ‘‘Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics,’’ Appl. Sci., vol. 11, no. 20, p. 9554, Oct. 2021.

[18] W. Liu, X. Deng, G. Huang, and A. Y. Fu, ‘‘An antiphishing strategy based on visual similarity assessment,’’ IEEE Internet Comput., vol. 10, no. 2, pp. 58–65, Mar. 2006.

[19] A. Rusu and V. Govindaraju, ‘‘Visual CAPTCHA with handwritten image analysis,’’ in Proc. Int. Workshop Human Interact. Proofs. Berlin, Germany: Springer, 2005, pp. 42–52.

Author Profiles

SK. ANJANEYULU BABU is an Associate Professor in the Department of Master of Computer Applications at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. His Specilization AI&ML. He is committed to advancing research and fostering innovation while mentoring students to excel in both academic and professional pursuits.

Mr. P. BRAHMAIAH is an MCA Student in the Department of Computer Application at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He has Completed Degree in MSCS (Statistics) from M.S.R Degree College Kavali, Nellore district.