ISSN: 2249-7196
IJMRR/Sep. 2024/ Volume 14/Issue 7/183-193
Mr. Praveen Kumar / International Journal of Management Research & Review

PHISHCATCHER: XGBOOST ALGORITHM-BASED CLIENT-SIDE PROTECTION AGAINST WEB SPOOFING ATTACKS

Mr. Praveen Kumar[1], B.Pavani[2], P.Parameshwari [3], V.Akhila[4],

[1] Assistant Professor, Department of CSE-AIML, Malla Reddy Engineering College for Women (Autonomous Institution), Hyderabad,

[2],[3],[4] Student, Department of CSE-AIML, Malla Reddy Engineering College for Women (Autonomous Institution), Hyderabad.

ABSTRACT: *The secrecy and authenticity of user data, including passwords and PIN codes, provide a significant problem for cyber security. Every day, phony login screens asking for private information are shown to billions of visitors. Many techniques, including phishing emails, alluring adverts, clickjacking, spyware, injection of SQL, session hijacking, man-in-the-middle attacks, denial of service, and cross-site scripting assaults, may be used to deceive a user into visiting a website. Web spoofing, often known as phishing, is an online scam where the perpetrator creates a fraudulent replica of an authentic website and demands personal data from users, including passwords. Researchers have suggested a number of security measures to thwart these vulnerabilities, however all have problems with precision and latency. We suggest and create a client-side defense mechanism that utilizes machine learning methods to identify spoof websites and shield users from phishing attempts in order to address these problems. We construct the machine learning algorithm to classify URLs as trustworthy or suspicious, and as a proof of concept, we create the Phish Catcher Google Chrome plugin. The program determines if a login website is spoof or not based on four distinct kinds of web properties that are loaded into a random forest classifier. Several studies were conducted on actual web apps to evaluate the extension's precision and accuracy. Based on 400 phished and 400 real URLs, the experiment's findings demonstrate an impressive 98.5% accuracy and 98.5% precision. In addition, we conducted tests on forty phished URLs in order to gauge the lag of our tool. Phish Catcher's reported reaction time was under 62.5 milliseconds on average.*

INTRODUCTION

The National Institute für Research in Digital Sciences and Technologies (Inria) users and members in France got an email in French in October 2022 requesting them to validate their webmail accounts using the provided direct link.

At https://www.educationonline.nl/Cliquez.ici.cas.inria.fr.cas.login/login.html, you can easily log in. The assistant editor who oversaw this manuscript's assessment and gave it the go-ahead for publishing was Seifedine Kadry; nevertheless, when you click on the link, it leads to a phony website that looks real. 1.On October 10, 2022, users of the Coq-club Inria https://www.inria.fr/en got an email alerting them to the phishing assault. This email was sent via the Inria central authentication login page. Given how well this fictitious login page mimics the Inria official login page at https://cas.inria.fr/cas/login?service=users will inadvertently input their Inria username and password on a phony website, which the attacker may then submit to the authentic Inria login page. This is a fraud attempt against Inria and its registered users/members. Here are links to the official and phony Inria login sites. Because both websites are identical, people may easily become victims of this phishing scheme. We have tested Phish Catcher, our tool, against these and a few additional assaults that are described in Section V. Phish Catcher: Protecting Client-Side against Internet Spoofing Attacks FIGURE 1. An attempt at phishing by Inria. The online world has greatly expanded due to the amazing advancements in current technology, including e-commerce, e-banking, e-health, e-governance, and remote learning. Since social networking sites like Facebook and Twitter are playing a major part in the present era's globalization, billions of users have embraced this growing trend. Users of many websites have the option to register for an account in order to have a personalized experience. Users must set up a customized account in order to use the websites' online specialist services. For this reason, visitors are often directed to login webpages where they must create and register an identity (such as a username) and secret (such as a password) in order to set up an account. The user submits a web request the next time they need to access a distant resource or service, and they get a login form to fill out with their identity and secret. The privacy of users is now very vulnerable to identity theft and the loss of private information. As shown in Figure 2, a phishing assault scenario starts with an email that contains a link to a rogue website. There may be content in the email that entices or persuades the recipient to click and follow the pointer. The

page seems authentic as the trustworthy website where the user has an account when the unwary user clicks and opens it. The attacker receives the affected user's confidential information once they submit it by pressing the submit or log in button and entering their username and password. After launching the phishing assault, the attacker obtains the hidden credentials and uses them to get in to the authentic website. The prevalence of identity theft, online fraud, and scams has significantly grown with the introduction of phishing and web spoofing tactics. Phishing, often known as web spoofing, is a kind of cybercrime when a malevolent hacker attempts to get sensitive information from the victim. To endanger online systems, attackers have used a variety of phishing & web spoofing tactics. Web spoofing was first employed for identity theft, but these days, hackers utilize it to get private data pertaining to Americans. an ordinary phishing assault. organization secrets, security, and intellectual property. Phishing assaults in the modern day have already evolved into new areas, such as spear phishing, faking mobile applications, and phishing using QR codes, among others. Scam tactics and assaults like this have the potential to get past defenses like digital certificates, firewalls, encryption software, and two factor authentication. These two-factor authentication solutions are widely used by businesses as a safeguard against identity theft and financial frauds. Sadly, all of these systems are now susceptible to sophisticated fraud techniques. Volume 11, 2023, 61250 Phish Catcher: Client-Side Protection Against Web Spoofing Attacks, by M. Ahmed et al. Logos taken from an honest site are often used by the attackers on their spoof websites to mimic their look, either by keeping copies of the logos or putting links to them. The attacker may use HTML from the truthful website along with logos, altering it as needed. The attackers utilize email, trojan horses, key loggers, and man-in-the-middle proxies as phishing attack routes to fool users. Online banking websites, e-commerce websites, and third-party payment systems are the most often attacked industrial sectors by cybercriminals. Because phisers target non-cryptographic components, SSL/TLS cryptographic security techniques are insufficient to fully protect against them. In order to withstand fraudulent attempts, these protocols need to be reinforced with extra security measures. Either the client-side, the server-side, or both may enforce these techniques. The majority of developers often overlook the laborious task of altering websites in order to include server-side solutions. On one hand, client-side solutions shield users without requiring server assistance. While client-side solutions are the main emphasis of this research, server-side solutions could also be useful in

detecting spoofing websites. The majority of anti-spoofing methods rely on URLs, passwords, or third-party certifications. Stateful and stateless anti-spoofing technologies are occasionally distinguished from one another. They might also be categorized using heuristics and blacklists, which are automated phishing detection mechanisms. Black/white list-based tools miss zero-day assaults yet produce almost no false alarms (accuracy) and are able to identify about 90% of phishing sites. Moreover, black-listing techniques have a number of shortcomings, including the inability to stop newly launched assaults and changing domains, as well as an increased susceptibility to spam URLs. The results of using heuristic-based strategies to catch phish sites that are not on black lists are highly promising. 90% of phishing sites can be identified using heuristic (content)-based tools like CANTINA and Spoof Catch, with just 1% of false positives. The tool Fake Catch has a latency that is measured in seconds, and it becomes worse with time. Although stateful anti-phish algorithms have outstanding accuracy, their performance deteriorates with time since they rapidly fill up local storage. A handful of login page photos are first compared for visual resemblance in Spoof Catch, but when the user browses more websites, the amount of password page images grows within the local storage. Furthermore, this lengthens the time it takes to match each login picture in the storage with the picture of a login page that was received. In keeping with this line of inquiry, we build and implement a machine learning (ML)-based stateless anti-phishing solution. In the last ten years, several distinguished scholars have suggested machine learning methods for identifying malevolent URLs in order to prevent future scams. ML techniques use large collections of URLs as training data. Whether a requested URL is scam-free or not is suggested based on the statistical features acquired by the training sets. The main issue with utilizing ML for URL identification is training data. After training data is acquired, it undergoes further processing to produce a mathematical model. Since simple words may not be able to predict the state for the URL under test, gathering characteristics form the training data is the main priority. Finally, a projected version from the initial data yields an actual model. Many academics employ machine learning approaches, such as Naive Bayes, supported vector machines (SVM), and logarithmic regression (LR), for this purpose; nevertheless, these algorithms are subject to a number of difficulties. In order to defend against online spoofing attacks, we suggest and create a stateless the client's side tool called Phish Catcher in this work. Based on artificial intelligence methods, the Phish Catcher Google Chrome

plugin uses an algorithm called random forest to determine if a login page is authentic or fake. We tested the Phish Catcher's accuracy and efficiency on actual online apps, and results were astounding.

RELATED WORK

Spoof Catch: A client-side protection tool against phishing attacks

The majority of ant phishing methods in the literature both suffer from both issues or evade certain attack patterns in order to defend against online spoofing assaults. These solutions are built on intricate parameter setups. In this paper, we suggest that the user's entire visual experience of the website may be used to thwart phishing assaults. Our claim may be realized by implementing a client-side security method that we propose, called Spoof Catch, based on the visual resemblance of web pages. Four algorithms have been built and added into the extension to compare the similarity of real and phished web pages. Numerous, in-depth tests have been carried out to assess the solution, showing that Fake Catch can effectively intercept all phishing assaults with a manageable overhead.

A framework for detection and measurement of phishing attacks

Phishing is a kind of identity theft whereby advanced attack vectors and social engineering tactics are used to steal financial information from unwary customers. A phisher will often attempt to trick her target into clicking on a URL that leads to a malicious website. In this research, we primarily investigate the URL structures used in different types of phishing assaults. It may often be determined if a URL is associated with a phishing attempt without necessitating an understanding of the associated page data. We go over a few characteristics that may be used to tell a legitimate URL from a phishing one. A the logistic regression method filter with good accuracy and efficiency is modeled using these characteristics. We assess the frequency of fraud on the Internet nowadays by using this filter to undertake extensive measurements on a few billion URLs.

Effective protection against phishing and web spoofing

On the Internet, phishing and Internet spoofing have become quite common and annoying. The attacks are challenging to defend against primarily because they go at non-cryptographic elements like user input or the user experience of a web browser. This indicates that extra

security measures must be added to cryptographic safety protocols, such the SSL/TLS protocol, in order to fully combat the assaults. In this work, we provide an overview, analyze, and assessment of these defenses against (massive) phishing & Web spoofing assaults.

Defending against injection attacks through context-sensitive string evaluation

An important danger to application-level security is injection vulnerabilities. Shell injection, SQL injection, and cross-site scripting vulnerabilities are a few of the most prevalent kinds. As a result of their heavy reliance on application developers, current approaches for thwarting injection attacks—attacks that take use of these vulnerabilities—are prone to mistake. In this research, we provide a technique to identify and stop injection attacks called CSSE. The way CSSE works is by tackling the ad-hoc encoding of user-provided information, which is the main reason these kinds of assaults may be successful. By combining metadata assignment to user-provided input, data-preserving string operations, and context-sensitive string evaluation, it offers a platform-enforced channel separation. No changes to the application source code or communication with application developers are necessary while using CSSE. It efficiently transfers the responsibility of building safeguards for attacks involving injection from the many developers of applications to the tiny group of security-savvy platform engineers, as only modifications on the foundational platform are required. In addition to being more successful against the majority of injection attack types, our approach also has a lower mistake rate than previous methods put out. We have created a prototype version of CSSE for PHP, a technology that is especially vulnerable to these flaws. We validated our approach using phpBB, a popular bulletin-board program, with our prototype. With just a little amount of run-time overhead, CSSE was able to identify and stop every SQL injection attempt that we were able to replicate.

Reliable protection against session fixation attacks

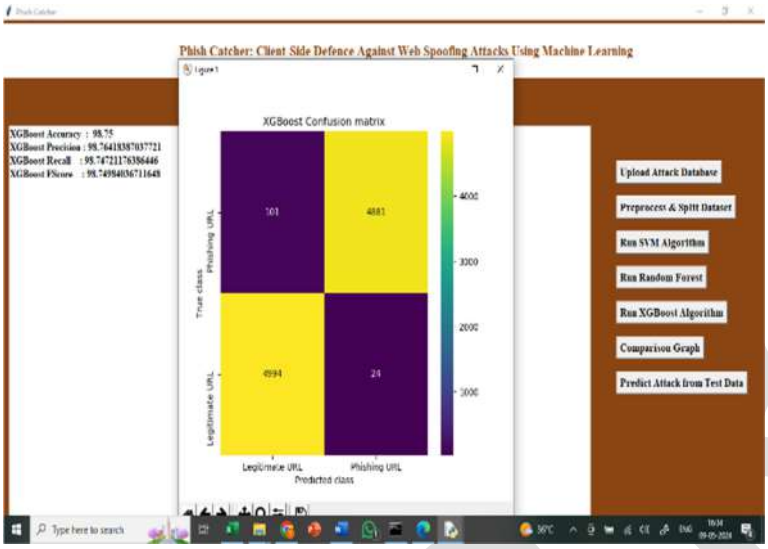
Issues with Web applications that, in some situations, allow the adversary to carry out a Session Hijacking attack by taking control of the victim's jet identifier value are referred to as "session fixation vulnerabilities." The attacker may completely assume the victim's identity and target the weak Web application if the attack is successful. We analyze the pattern of vulnerabilities and pinpoint their underlying cause: the application's logic, which manages authentication procedures, and the structure of support, which takes care of session tracking, have different concerns. In light of this finding, we outline and go over three different server-side solutions to

mitigate Session Fixation issues. Every countermeasure we have developed is customized to fit a certain real-world situation that an operator of a weak Web service may run across.

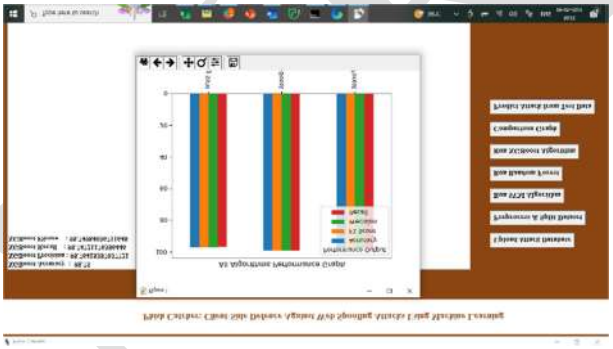
METHODOLOGY

1. 1**.UploadDataset**: The Attack dataset can be uploaded to the program using this module.
2. **Preprocess & Split Dataset:** This module allows us to use processing strategies including normalization, separating data into test and training sets, and shuffling.
3. **Run SVM Algorithm:** With the help of this module, the SVM training achieved 96% accuracy. Additionally, the confusion matrix graph displayed measures such as precision, FSCORE, recall, and others.
4. **Run Random Forest Algorithm:** Using this module to train the Random Forest algorithm, it achieved 98% accuracy and displayed metrics such as precision, recall, FSCORE, and confusion matrix graph.
5. **Run Xgboost Algorithm:** Using this With 99% accuracy, the module training extensions XGBOOST technique may also be shown in a confusion matrix graph along with other metrics including precision, recall, and FSCORE.
6. **Comparison Graph:** This module shows a graph of the performance of every algorithm.
7. **Predict Attack From Test Data:** By reading TEST URLS using test data using this module and then using the XGBOOST extension, we can determine if a URL is phishing or saved.

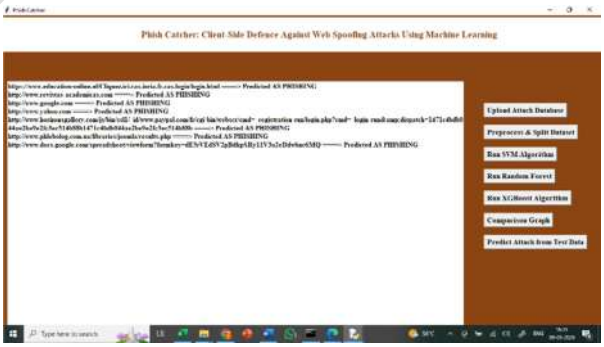
RESULT AND DISCUSSION



The XGBOOST algorithm achieved 99% accuracy in the training extension shown above.



The algorithm names are shown on the x-axis by the algorithm names, accuracy and additional metrics are represented on the y-axis by various color bars, and XGBOOST achieved good accuracy across all algorithm extensions in the graph above.



In above screen defining code to read TEST URLS from test data and then using extension XGBOOST we are predicting weather URL is save or PHISHING and after executing this block will get output.

In above screen before arrow =➔ symbol we can see TEST URL and after =➔ arrow symbol we can see predicted output as 'SAFE or PHISHING'

CONCLUSION

These days, a lot of our information comes from the internet, including emails, posts, reviews, online news, and much more. Attackers may now entice regular users with phony phishing URLs or spoof websites by offering them attractive messages such as winning a jackpot thanks to this internet access. When a user clicks on such a URL or visits a fake website, a popup window appears asking the user to input their login credentials. From there, attackers may use those credentials to access banking or other financial websites, seize or steal the user's whole balance or other confidential data.

The author of this research uses the Random Forest method to identify phishing URLs since many machine learning and signature-based techniques that have been published to date have an inaccurate detection rate and are thus not suitable for avoiding such URLs. To improve prediction accuracy, the Random Forest approach incorporates support for feature selection and tuning. Using a collection of trees, Random Forest will filter and eliminate unnecessary data from the dataset, leaving just optimum features.

You may read a lot more information from the author in the base article. We may predict if a URL is safe or phishing by utilizing the PHISHTANK dataset, which has thousands of legitimate and phishing URLs, which was utilized by the author to train their proposed algorithm. In addition to training, the author has created a Chrome plugin that analyzes every URL a user visits and notifies them of SAFE or phishing URLs. An existing SVM method will be contrasted with the proposed Random Forest technique.

REFERENCES

[1] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, ''spoof Catch: 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. VOLUME 11, 2023 61261 M. Ahmed et al.: PhishCatcher: Client-Side Defense Against Web Spoofing Attacks

[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.