WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Detection and Prevention of Phishing Using Machine Learning

Harshal Mali[a], Parth Chavan[a], Aryan Habib[a], Aditya Dhotre[a], Naresh Kamble[b]

Student,Sanjay Ghodawat Polytechnic, Atigre,India
Faculty of Sanjay Ghodawat Polytechnic (CSE Dept.),Atigre,India

A B S T R A C T

The back-end of a page is completely unknown to naive browser users. Users may be duped into revealing their passwords or downloading harmful material. Our goal is to design a Chrome plugin on proposed project that would function as a middleman between users and harmful websites, reducing the possibility of users succumbing to them. Furthermore, because all hazardous content is constantly evolving, it is impossible to collect it all. To combat this, we're utilising machine learning to train the tool and categorise the fresh content it sees each time into certain categories, allowing us to take appropriate action. Malicious web pages contain content that can be utilised by attackers to take advantage of end-users. This contains phishing URLs, spam URLs, JavaScript malware scripts, Adware, and a variety of other things. The purpose is to offer safe browsing regardless of which website the user wants to visit. Even if the user visits a phishing website, precautions will be taken to ensure that the user is not damaged.

Keywords:Machine learning, Chrome, Extension, Phishing, Malicious, URL.

1. INTRODUCTION

The Malicious web pages contain content that can be utilized by attackers to take advantage of end-users. This contains phishing URLs, spam URLs, JavaScript malware scripts, Adware, and a variety of other things. Due to the constant development of new strategies for carrying out such attacks, it is becoming increasingly difficult to detect such vulnerabilities. Furthermore, not all users are aware of the many types of exploits that might be used by attackers. As a result, if a user is unaware of a vulnerability in a web page, this tool will assist him in remaining safe despite his lack of expertise of the website. Furthermore, if the URL appears to be a phishing URL, the user will be prevented from that website. Python, because it is open source and has a wide range of support libraries, simple syntax, and a wealth of resources, is the best choice for implementing machine learning. The alternative method is to look up the entered URL in a list of websites that have been declared malicious by a reliable source. The disadvantage of this method is that the list is not exhaustive, i.e. it grows every day. Furthermore, due to such a large list, the system's latency time will always increase.

1.1. Objective of Project

- To detect and eliminate risk of phishing.
- To detect website spoofing.
- To overcome the drawbacks of existing solutions which fails to serve the purpose.
- To implement Real time URL classification using Phishtank

* *Corresponding author.* Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: author@institute.xxx

2. LITERATURE REVIEW

Researchers have used machine learning technologies to detect harmful URLs in the literature. Machine learning uses statistical attributes to learn a prediction model and classifies a URL as harmful or benign. To extract features, this approach analyses URLs and their corresponding websites or web page information. Static and dynamic features are two types of features that can be extracted using this method. Literature extracts lexical information from URL strings, host information, and occasionally HTML and JavaScript content. The support vector machine (SVM) is used to detect a variety of network traffic-related features extracted from URL in the literature. Three feature processing strategies have been proposed in the literature to improve the classification effect. While the approaches described above have demonstrated to be effective, they do have certain drawbacks. Traditional machine learning-based detection methods frequently necessitate manually extracting characteristics. By designing these features, attackers can escape being identified by current detection approaches, making it extremely difficult to maintain a detection system based on typical machine learning. Furthermore, when detecting fraudulent URLs on a broad scale, a trained model may lose some essential information from the URL.

2.1. Signature based Malicious URL Detection

Long ago, studies on malicious URL detection utilising signature sets were researched and implemented The vast majority of these investigations rely on databases of known harmful URLs. A database query is run whenever a new URL is accessed. If a URL is blacklisted, it is assumed to be dangerous, and a warning is issued; otherwise, URLs are assumed to be safe. The biggest drawback of this strategy is that it will be extremely difficult to detect new malicious URLs that are not included in the provided list.

2.2. Machine Learning based Malicious URL Detection

Supervised learning, unsupervised learning, and semisupervised learning are the three types of machine learning algorithms that can be used to detect dangerous URLs. The methods of detection are based on URL behaviour. In, researchers looked into a number of harmful URL systems based on machine learning techniques. SVM, Logistic Regression, Nave Bayes, Decision Trees, Ensembles, Online Learning, and so on are examples of machine learning algorithms.

2.3. Malicious URL Detection Tools

- URL Void: URL Void is a URL checking program using multiple engines and blacklists of domains. Some examples of URL Void are Google SafeBrowsing, Norton SafeWeb and MyWOT.
- UnMask Parasites: Unmask Parasites is a URL testing tool by downloading provided links, parsing Hypertext Markup Language (HTML) codes, especially external links, iframes and JavaScript.
- Comodo Site Inspector: This is a malware and security hole detection tool. This helps users check URLs or enables webmasters to set up daily checks by downloading all the specified sites. and run them in a sandbox browser environment.
- Some other tools: Among aforementioned typical tools, there are some other URL checking tools, such as UnShorten.it, VirusTotal, Norton Safe Web, SiteAdvisor (by McAfee), Sucuri, Browser Defender, Net-craft, Online Link Scan, and Google Safe Browsing Diagnostic.

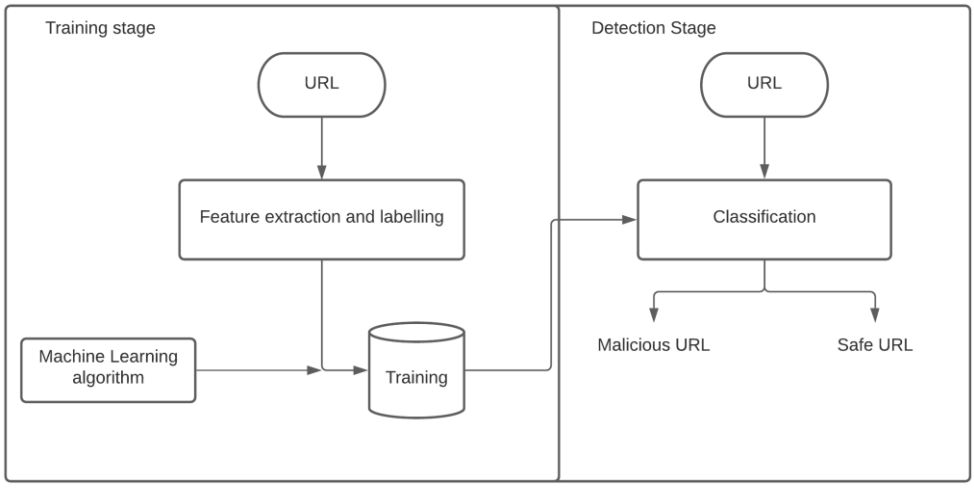

Fig. 1 - Malicious URL Detection Model using Machine Learning.

3. METHODOLOGY

3.1. Malicious URL Detection Tools:

Machine learning works by analysing what features are extracted and the URL is tested on the classifier. To train the classifier, we used the UCI Phishing Website Dataset https://archive.ics.uci.edu/ml/datasets.php.

3.1.1. Obtaining Dataset :

The dataset was obtained from the UCI - Machine Learning Repository which contains the Phishing Web Site Dataset. This dataset is composed of 11055 entries of websites which are classified as phishing and benign. These entries each have 30 features of the website used.

3.1.2. Feature Selection :

From the dataset, out of the 30 features present, it was infeasible to extract all the features. This is because many features used some standard databases which are not accessible to us. Also, extracting some of the features seemed not possible as they demanded the extraction of data from the server of the website, which is not possible. Hence, we narrowed down our dataset to contain 22 features.

3.1.3. Choosing Classification Algorithm :

For classifying the URL entered, as either safe or malicious, we considered the following algorithms:

1) The K-Nearest Neighbors (kNN) algorithm can be used to solve both classification and regression problems. However, it is mostly employed to solve categorization difficulties. The number of nearest neighbours we want to vote from is represented by the letter 'k' in the kNN method. This algorithm will search the training dataset for the closest k-samples when predicting for a new data sample.
2) SVMs (Support Vector Machines) are supervised learning models that can be used for classification and regression. The SVM algorithm works with a dataset that has input samples separated into two classes, each with a label of 0 or 1. Finding a line or a plane, also known as a hyperplane, that will most efficiently split the two classes is part of the process.

3.2. Phishtank

Phishing blacklists are a common defence tactic that aims to protect consumers against phishing attempts. These blacklists often comprise known phishing URLs, giving an access control list that is used to restrict people from visiting these risky websites. Google Safe Browsing (GSB), PhishTank (PT) https://phishtank.org/, and OpenPhish are three common phishing blacklists used nowadays (OP). These three blacklists are utilised by the web browsers Chrome, Safari, Firefox, and Opera, the email provider Yahoo! Mail, the antivirus companies McAfee, Kaspersky, Virus Total, and Strong Arm, and the online reputation and internet safety service web browser plugin Web Of Trust.

4. IMPLEMENTATION OF PROPOSED PROJECT

One of the most difficult aspects of our research was the paucity of phishing datasets. Despite the fact that several scientific publications on phishing detection have been published, none of them have given the dataset that they utilised in their research. [8]Another aspect that makes it difficult to discover an acceptable dataset is the lack of a common feature set for recording features of a phishing website. Some scholars thoroughly investigated and benchmarked the dataset we utilised in our study. Fortunately, the dataset's associated wiki includes a data description paper that details the data production methodologies used by the dataset's developers phishing-dataset. We have also incorporated code that pulls features of new phishing websites published by the PhishTank website in order to update our dataset with new phishing websites. The dataset comprises around 11,000 sample websites.The dataset's associated wiki contains a data description paper that goes over the data production methodologies used.

1) Having IP Address: If an IP address is used instead of the domain name in the URL, such as "http://217.102.24.235/sample.html".

2) URL Length: Phishers can use a long URL to hide the doubtful part in the address bar.

3) Shortening Service: Links to the webpage that has a long URL. For example, the URL "http://sharif.hud.ac.uk/" can be shortened to "bit.ly/1sSEGTB".

4) Having @ Symbol: Using the "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol

5) Double Slash Redirection: The existence of "//" within the URL which means that the user will be redirected to another website

6) Prefix Suffix: Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example http://www.Confirme-paypal.com.

7) Having Sub Domain: Having subdomain in URL.

8) SSL State: Shows that website use SSL

9) Domain Registration Length: Based on the fact that a phishing website lives for a short period

10) Favicon: A favicon is a graphic image (icon) associated with a specific webpage. If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt.

11) Using Non-Standard Port: To control intrusions, it is much better to merely open ports that you need. Several firewalls, Proxy and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only open the ones selected

12) HTTPS token: Having deceiving "https" token in URL. For example, http://https-www-mellat-phish.ir

13) Request URL: Request URL examines whether the external objects contained within a webpage such as images, videos, and sounds are loaded from another domain.

14) URL of Anchor: An anchor is an element defined by the < a > tag. This feature is treated exactly as "Request URL".

15) Links In Tags: It is common for legitimate websites to use ¡Meta¿ tags to offer metadata about the HTML document; ¡Script¿ tags to create a client side script; and ¡Link¿ tags to retrieve other web resources.

16) Server Form Handler: If the domain name in SFHs is different from the domain name of the webpage.

17) Submitting Information To E-mail: A phisher might redirect the user's information to his email.

18) Abnormal URL: It is extracted from the WHOIS database. For a legitimate website, identity is typically part of its URL.

19) Website Redirect Count: If the redirection is more than four-time

20) Status Bar Customization: Use JavaScript to show a fake URL in the status bar to users

21) Disabling Right Click: It is treated exactly as "Using onMouseOver to hide the Link"

22) Using Pop-up Window: Showing having popo-up windows on the webpage.

23) IFrame: IFrame is an HTML tag used to display an additional webpage into one that is currently shown.

24) Age of Domain: If the age of the domain is less than a month.

25) DNS Record: Having the DNS record

26) Web Traffic: This feature measures the popularity of the website by determining the number of visitors.

27) Page Rank: Page rank is a value ranging from "0"to "1". PageRank aims to measure how important a webpage is on the Interne

28) Google Index: This feature examines whether a website is in Google's index or not

29) Links Pointing To Page: The number of links pointing to the web page.

30) Statistical Report: If the IP belongs to top phishing IPs or not.

5. CONCLUSION AND FUTURE WORK

A method for detecting malicious URLs using machine learning is presented in this paper. We do not use special attributes in this study, nor do we seek to create massive datasets to improve the system's accuracy, as many other traditional publications do. In this case, the combination of easy-to-calculate attributes and big data processing technologies to ensure the balance of the two factors is the system's processing time and accuracy. The findings of this study can be applied and implemented in information security technologies and systems. By combining this with a real-time phishing URL database, we will ensure the highest level of security for our users. It is worth noting that the combination of multiple classifiers does not always outperform the best individual classifier in the ensemble classifiers. The findings encourage future research to add more features to the dataset, which could improve the performance of these models; thus, it could combine machine learning models with other phishing detection techniques, such as List-Base methods, to achieve better performance. In addition, we will investigate the possibility of proposing and developing a new mechanism for extracting new features from the website in order to keep up with new phishing attack techniques.

REFERENCES

[1] R. S. Rao and S. T. Ali, "A computer vision technique to detect phishing attacks," in 2015 Fifth International Conference on Communication Systems and Network Technologies, pp. 596–601, IEEE, 2015.

2] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," Machine Learning and Knowledge Extraction, vol. 3, no. 3, pp. 672–694, 2021.

[3] R. C. Dodge Jr, C. Carver, and A. J. Ferguson, "Phishing for user security awareness," computers & security, vol. 26, no. 1, pp. 73–80, 2007.

[4] A. Altaher, "Phishing websites classification using hybrid svm and knn approach," International Journal of Advanced Computer Science and Applications, 2017.

[5] H. D. N. Cho Do Xuan, "Malicious url detection based on machine learning," International Journal of Advanced Computer Science and Applications, 2020.

[6] M. Canham, C. Posey, D. Strickland, and M. Constantino, "Phishing for long tails: Examining organizational repeat clickers and protective stewards," SAGE Open, vol. 11, no. 1, p. 2158244021990656, 2021.

[7] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 20–39, Springer, 2017.

[8] A. P. Rosiello, E. Kirda, F. Ferrandi, et al., "A layout-similarity-based approach for detecting phishing pages," in 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007, pp. 454–463, IEEE, 2017.