IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Phishing Detection Using Machine Learning

Sanskruti Kadam[1], Ajit Gaikwad[2], Atharva Kale[3]

Dr .Mohammad Khaja Shaik

*1,*2,*3, Student, Department of Computer Science, ISBM College of Engineering, Pune, Maharashtra, India

*4 Professor, Department of Computer Science, ISBM College of Engineering, Pune, Maharashtra, India

Abstract

In this paper, we propose a feature-free method to detect phishing websites using Normalized Compressed Distance (NCD), a parameter-free similarity measure that calculates the similarity between two websites with compression power, eliminating the need to perform some features. extraction. It also eliminates dependency on the features of a particular website. This method examines the HTML of web pages and calculates their similarity to known phishing websites to identify them. We perform phishing archetype extraction by using a cutting-edge algorithm to select samples that represent clusters of phishing websites. We also introduce the use of incremental learning algorithms as a framework for continuous updating without extracting new features when concept drift occurs. On large datasets, our proposed method outperforms previous methods in detecting phishing websites with an AUC score of 98.68% and a positive rate (TPR) of approximately 90%, while maintaining a low interest rate of 0.58%. Maintains rate (FPR). Our method uses a standard design, eliminates the need for long-term data storage in the future, and can be used in a real system with a processing time of approximately 0.3 seconds.

Index Terms- *HTML analysis, incremental learning algorithms, Normalized compressed distance, phishing archetypes extraction, Phishing detection.*

INTRODUCTION

Phishing is defined as a cyber attack that uses social engineering through digital means to persuade victims to reveal personal information such as their passwor ds or credit card numbers. Strategies used in phishing attacks exploit human vulnerability to distinguish between real and phishing messages or websites. Phishi ng is a lowcost but important tool that can aid in many cyber attacks as it is often used as a key step in regular threats. As our dependence on multiple digital pla tforms increases, phishing has become a versatile weapon in the attackers' arsenal. Although phishing has broad definitions, the term itself is often associated w ith phishing attacks that use email or text messages as an attack vector to trick victims into submitting private information on phishing websites or downloading malware. These websites are often carefully designed to look professional and trustworthy as if they were legitimate.

Research

There has been an increase in certain phishing attacks. These attacks have caused major financial losses estimated to be between $60 million and $3 billion per year in the United States . In another report, APWG detected approximately 65,400 phishing websites per month in 2018, while PhishLabs reported a 40.9% inc rease in phishing volume in 2018 compared to the previous year. PhishLabs also reports that the volume of attacks continues to increase as attackers adapt their methods and adapt to changes in the digital environment. Additionally, the use of free service providers has increased the number of phishing attacks over the p ast four years, from 3.0% in 2015 to 13.8% in 2018. Setting up a phishing website is also easy using phishing tools. The availability of these tools allows an act or to create multiple professionallooking phishing websites in a short time. For example, PhishLabs reported an increase in the number of attackers in August 2 018. As malicious organizations introduce these phishing devices, the number of phishing attacks is likely to increase in the future. The use of free hosting, phis hing devices and SSL certificates shows that attackers are constantly trying to use new methods and the number of phishing attacks continues to increase over t he years. This makes it difficult to develop a reliable phishing method that includes a good attack scenario.

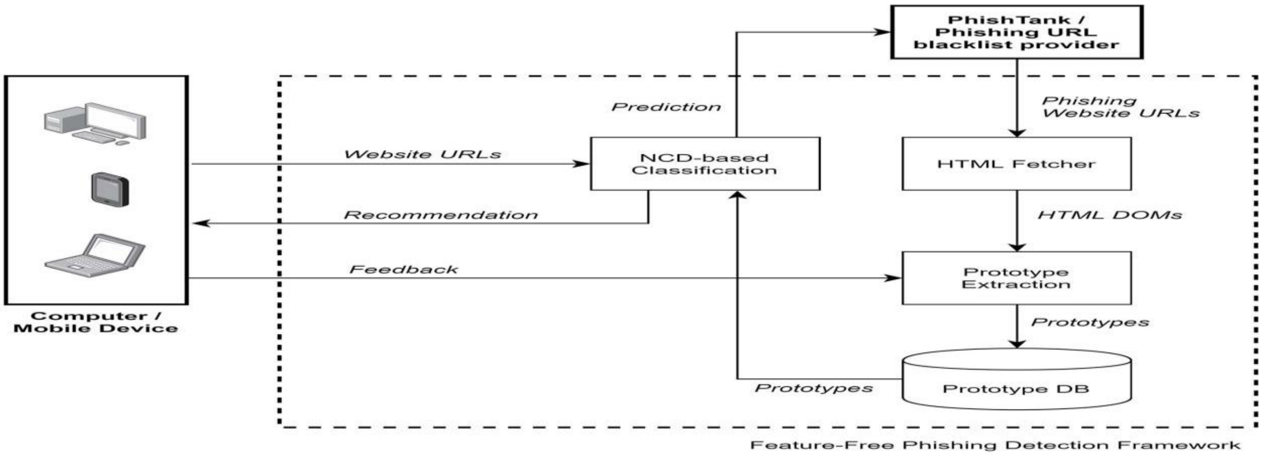
Solution

To guard against the dangers of phishing, researchers have learned in recent years many ways to create phishing machines that automatically detect websites by analyzing web content, images, URLs, and other network interactions. In general, this process can be divided into two groups. The first method discovers the basic characteristics of phishing websites and tries to detect these attacks based on certain charac teristics. In recent years, many studies have adopted this approach using machine learning and deep learning. While these techniques are effective at detecting p hishing, their detection is less resilient to conceptual drift because they often rely on features expected to be associated with phishing websites (e.g., specific typ es of websites or unusual patterns in URLs) that are future and irrelevant it will become. Meanwhile, the second method attempts to detect phishing by assessin g the similarity between the phishing website and the legitimate target website. This method is less vulnerable to zero day phishing attacks than the first method. But similar methods can quickly filter out many phishing websites before feeding them to machine learning processe s, which often require more time to classify. Most previous studies have suggested using a variety of similar metrics and models to identify similar phishing we bsites. These similar methods usually require modeling the website into a representation using DOM trees, bag language or Doc2Vec model. A free method that uses link averaging to calculate HTML web similarity. The rationale for this study is based on the work of Cuietal. Demonstrating the diversity of phishing web sites, shows that 90% of the 19,066 phishing websites identified by PhishTank were copies or modifications of other known phishing websites. Cui et al. The new distance metric, an equal measure, aims to measure the website's similarity t o a phishing search. This distance measure takes into account the presence of a predefined set of HTML tags that provide important information about whether a website is vulnerable to phishing attacks. On the other hand, the use of standard and static in our scheme will eliminate the need for predefined HTML text an d use the compression algorithm to universally measure two data streams according to the value of similar information contained in them. Our plan is not limite d to any specific type of phishing attack or phishing campaign, and is not limited to any type of phishing email. However, our approach is limited to the evoluti on of phishing websites that have appeared at least once and cannot identify new phishing websites with different and unique HTML patterns. A simple method is presented to perform a web similarity index to identify similar phishing websites using difference in differences (NCD). The relationship between content an d perspective and what path the content follows

Concept and definitions

In this section, we introduce and discuss the use of Normalized Compressed Distance (NCD) to perform phishing website detection and model-based learning algorithm for clustering and distributed websites. NCD to measure phishing website similarity NCD is a free index that is universal, so it tries to estimate the similarity of key features in each profile or product comparison. The purpose of NCD is to capture all distances, including valid Hamming distance, Euclidean distance, and correction distance standards. Appendix A provides more details about NCDs. Based on these characteristics, we set out to investigate whether the use of NCD as a measure of dissimilarity is suitable for online phishing investigations. Due to the dynamic natur e of phishing, detection systems that rely on a given set of static features may fail to detect phishing when attack behavior changes. It is the product or information se lected for comparison in phishing detection. Chenetal. used NCD to measure the visual similarity between websites and detect phishing websites by calculating the N CD of screenshots of two websites. Their research sought to detect phishing attacks in cases where the phishing website was highly visible to the target's legitimate w ebsite. Chen et al. Although there are some minor differences between phishing and legitimate websites, it is believed that attackers must create phishing pages to loo k like legitimate pages to ensure that website users end up with the legitimate site. Based on this theory, they made a phishing website by calculating the NCD of sus picious websites and legitimate websites. An NCD value lower than a threshold indicates that the website is pretending to be a legitimate website and is therefore cla ssified as a phishing website. While this assumption may be true for some phishing websites, we found that in most cases the phishing website is not the same as the t arget website found that 90% of phishing files collected in 10 months of 2016 were variations or copies of other previous attacks in the database, indicating the common stability a nd quality of phishing websites. This is understandable given the increasing use of phishing devices, which is leading to the emergence of HTML-like content on new phishing sites. Therefore, we chose a new way to conduct phishing research by detecting similarities in the HTML content of websites, since phis hing websites are often created with special templates or equipment. Therefore, in this study, we perform binary NCD calculations on website HTML datasets to eval uate website similarity and detect phishing websites. More details about the design will be discussed in the next section. Learning Based Learning Using the NCD metric to measure the similarity between two websites, we cluster phishing websites with similar HTML content to classify them into groups and cla ssify the websites into groups of previous websites

with similar features . Our target group is a similar group represented by a small number of NCD outcomes. or features that make it clear and stable. First, we aim to make the system non-specific, meaning that it can be learned directly from the material without manual work. Additionally, one of the advantages of not using signatures is that the system can be adjusted to changes in phishing behavior or data presentation. Second, we try to create a search that can continue to learn and gain knowledge from the previo us learning process and combine this with new information obtained during the processing of data to create a work in progress. Thanks to the ability to obtain more i nformation, the system aims to improve detection time when encountering new phishing samples.



. **PHISHSIM Overview**

We offer PhishSim, a server-based phishing detection system that businesses can use across their customers' intranets, Internet Service Providers (ISPs), and cloud providers such as Amazon, Mi crosoft, and Google. Against phishing. Figure 3 shows at a high level how the system works and how it is used. The system takes as input the website URL requested by the user. It then gives a recommendation as to whether the website is safe or malicious. The proposed output is produced by NCD-based classifiers using samples stored in the Prototype DB database. The system can also update the database model by accepting new phishing names and creating n ew models. New phishing websites can be obtained from user reports and feedback and phishing blacklist databases such as PhishTank. Two main topics: Phishing w ebsite classification and phishing pattern database update. Phishing Site Classification

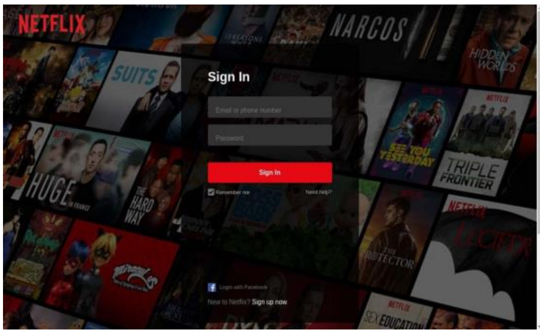
To perform phishing detection, our application retrieves the website URL when the user is about to open the HTML Document Object Model (DOM) website, thus si mulating how the page text would be displayed through a web browser. To retrieve the HTML DOM of the website, we use Chromium , an open source software proj ect that forms the basis of many websites, including Google Chrome and Microsoft Edge. Fig. 3. System diagram of stealth phishing signature PhishSim. The reverse classification system predicts whether a website is phishing or legitimate. If the website is suspected to be a phishing site, the result will be ad ded to the storage or blacklist and the user will be redirected to a warning page when they access the web page (it is recommended that you do not open the page). Ou r system can be used with Google Safe Browsing , which is based on a list of URLs of websites containing phishing content. Previously, as HTML messages were ad ded to the content, only the HTML tags were left rendered and displayed in the browser. Therefore, adding invisible elements to HTML will not affect performance.

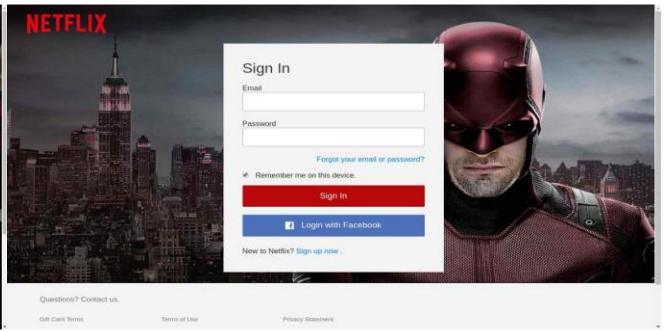
Phishing prototype database update

In order to maintain prediction accuracy, our system has a plan to update the phishing prototype database. The system is able to update the database model by regular ly (e.g. daily or weekly) retrieving data and user data from phishing blacklist providers. After receiving the HTML DOM of the web page, the system processes the te mplate to extract the template from this new file. Copies have been archived for NCD-based distribution. Similarity Analysis

To complete our approach, we conducted two experiments by sampling the number of NCDs to analyze the features and similarities between phishing websites and l egitimate websites. In the first experiment, we matched NCD calculations and combined them into specific datasets to analyze the similarity between phishing and le gitimate websites. The purpose of this test is to monitor the relationship between all websites, especially the relationship between websites belonging to different bra nds (phishing websites and legitimate websites). In the second test, we used the same method as the second test for different types of phishing and legitimate website information.

A. **Brand-Specific Similarity Analysis**





Phishing Websites with Similar HTML Contents (Cluster 1). Netflix Legitimate Website.

To evaluate the relationship between these websites based on their content, we calculate the NCD results and create a separate HTML DOM archive file. The mapped NCD values of HTML DOM profiles can be viewed as a cluster dendrogram as shown in Figure 4. Legitimate websites are paired with two phishing sites, while other phishing sit es are paired together. As shown in the dendrogram, the most similar phishing websites are P_NTF_52 and P_NTF_60, with NCD of 0.04. Interestingly, these two w ebsites have unique designs, as shown in Figure 6. The only difference is the code changes specified when loading the CSS stylesheet. Judging by the similarities in t he HTML DOM structure, it appears that these websites were created using the same phishing tools. We found that this was also true for other webgroups (P_NTF_2 1, P_NTF_35, P_NTF_2, P_NTF_5) and (P_NTF_37, P_NTF_22). Similar to the initial case of P_NTF_52 and P_NTF_60, the combined websites have the same H TML DOM structure but different website design. We also found that sometimes the website template is similar but the background image is different. We also found that the legitimate website was associated with one of the phishing websites, P_NTF_58. Although this may result in false positives in detection when using NCD-based similarity measures to detect phishing, we believe that this situation is rare and can be prevented by placing a Free registration on official websites. The purpos e of the whitelisting method is to filter out legitimate websites by comparing their content and writing with one of the whitelisted websites. Phishing websites using t his method will not meet the evasion requirements and will continue to pass similar NCD-based tests. binary NCD values. We group the screenshot image file using the same method as the Netflix HTML DOM file. View cluster dendrogram

A. Evaluation method

There are three experiments in this research. The first test is to evaluate the effectiveness of our model in detecting phishing websites and compare the res ults with other methods with equal difference to the method proposed by Cui et al. [18] and using the Doc2Vec model and Manhattan distance reported by Feng et al. [20]. In this experiment, we also evaluate the detection performance of various legitimate benchmarks for phishing groups and when using NC D in combination with other distance metrics. In the second experiment, we simulated additional detection of phishing websites and evaluated the perform ance of incremental learning models with NCD. Finally, in the third test, we analyzed the memory requirements and runtime performance to evaluate the f easibility of Phish Sim. In this study, we used the LZMA algorithm when calculating NCD values because it can better estimate the NID, as shown in Appendix B, resulting in better results and better phishing detection performance. Dataset To evaluate the performance, we use our own dataset, which is completely different from the dataset used in similar analysis (Section V) and has the best quality in distance selection (Section VI). We plan to provide experimental data for future research studies. To load phishing pages immediately. Phishing URLs received by Phish Tank are typically fake login or login pages or X page.

Conclusion

In this paper, we propose a feature-free method to detect phishing websites using Normalized Compressed Distance (NCD), which measures website similarity through compression without any f eature extraction or targeting some website features. This method examines the HTML source code of a web page and calculates its similarity to known phishin g websites. We also introduce the use of incremental learning algorithms as a framework for continuous change detection without having to repeat new feature extraction when concept drift occurs. Evaluating the performance of recent big data, our proposed method outperforms previous studies in detecting phishing w ebsites, achieving an AUC score of 98.68% and true quality around 90%, while maintaining a low FPR of 0.58%.

B. **Appendix A**

Normalized Compressed Distance

Normalized Compressed Distance (NCD) is an application-independent data theoretic method for measuring similarity between two objects. It is a unique tool that uses compression algorithms to perform data sepa ration and classification in various applications. In this way, useful information can be obtained from information without prior knowledge, as it works on universal information, regardless of its type, structure or view. To calculate the distance between two archives, compress both archives together and comp are with the results of compressing each archive separately. The main idea behind NCD is that combining two similar archives before compressing gives b etter results compared to the entire file size when compressed alone. On the other hand, compressing two archives that have little or nothing in common is less useful than compressing themselves. other products. This idea is expressed by the data distance E(x, y), which is defined as the length of the shortest binary program to compute y from x or compute x from y, which can be rewritten as E(x, ,) . y)) = max {K(x y), K(yx)} (4)

The information distance is based on the concept of Kolmogorov complexity K(x), which defines the length of the shortest program to compute x. > Expl ains how two objects are similar. Similar materials will give smaller NCD values, while special materials will give NCD values close to 1. Meanwhile, the top middle z is a small error caused by a flaw in the compression algorithm and is usually less than 0.1 for most standard algori thms [17].

For a given compressor, the C approximation of the NID denominator in Equation 5 is straightforward and gives us the NCD denominator in Equation 6. The number in Equation 5 can be expressed as: <br< b="" style="margin: 0px; padding: 0px;"></br<>>max {K(x, y)) ≤ K(x), K(x, y) ≤ K(y)} (7) and< K(x, y) = K(xy) = K(yx) (8) where xy or yx represents the union of x and y and K(x, y) is the shortest formula (x, y)' Represents the calculated length of . This approach is presented u sing K(x,y) because the union of xy is easier to compress. Equation 7 can best be approximated as min{C(xy), C(yx)} – min{C(x), C(y)} (9) C(x, y) was used in our experiment [17] To replace min{C(xy), C(yx)} as suggested in . Here we assume that C is symmetric, that is, C(xy) = C(yx).

Appendix B

C. Compression Algorithm Selection

D. In this section, we evaluate the performance of PhishSim using various compression algorithms (e.g. zlib, bz2, LZMA, and gzip). For comparison purposes, we used a different phishing technique and legitimate website data than that used in the main experiment. To generate the phishing data for this test, we used a list of 9,245 phishing URLs reported by users to PhishTank [9] between November 7, 2008 and March 28, 2020. We have collected non-phishing data as an intermediary collected from legitimate websites. related pages

References

[1] Alexa. Top 500 places in every country. Deadline: May 11, 2020. Current Status: https://www.alexa.com/topsites/countries

[2] Attackers use Morse code and other encryption methods in phishing evasion campaigns. Application Deadline: December 17, 2021. Available at: https://www.mi crosoft.com/security/blog/2021/08/12/attackers-use-morse-code-other-encryption-methods -in-evasive- phishing -campaign/

[3] Always log in. Visit time: March. March 25, 2021. Available: https://commoncrawl.org/

[4] Google Custom Search. Deadline: March 4, 2019. [online]. Available: https://developers.google.com/custom-search/

[5] Phisher likes: Five months later, Microsoft was replaced by PayPal. Deadline: April 12, 2020. Available at: https://www.vadesecure.com/en/phishers-favorites-q3-2019/

[6] Phishers' Favorites: It's not always easy to get high: Microsoft still top phishers Phishing is the number one type of counterfeit product. Deadline: January 28, 202 0. Available: https://www.vadesecure.com/en/phishers-favorites-q1-2019/

[7] Phishers' Favorites: Microsoft ranks first, but Facebook phishing is on the rise. Deadline: April 12, 2020. Available: https://www.vadesecure.com/en/phishers-favorites-q2-2019/