International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 9 Issue 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

Detection of Phishing Websites

Prof. Sumedha Ayachit[1], Pradnya Digole[2], Parth Chaudhari[3], Pratiksha Bhalerao[4], Madhuri Aradwad[5]

[1]Professor, Dept. Of Information Technology, JSCOE, Pune, Maharashtra, India
[2,3,4,5] Dept. Of Information Technology, JSCOE, Pune, Maharashtra, India
Department of Information Technology, Jayawantrao Sawant College of Engineering, Pune

---

Abstract –

The World Wide Web handles a large amount of data. The web doubles in size every six to ten months. World Wide Web helps anyone to download and download relevant data and important content for the website can be used in all fields. The website has become the main target of the attacker. Criminals are embedded in the content on web pages with the intent to commit atrocities. That audio content includes ads, as well as known and important user-friendly data. Whenever a user finds any information on a website and delivers audio content. Web mining is one of the mining technologies that create data with a large amount of web data to improve web service. Inexperienced users using the browser have no information about the domain of the page. Users may be tricked into giving out their personal information or downloading malicious data. Our goal is to create an extension that will act as middleware between users and malicious websites and reduce users' risk of allowing such websites. In addition, all harmful content cannot be fully collected as this is also a liability for further development. To counteract this, we use web crawling and break down the new content you see all the time into specific categories so that appropriate action can be taken. The problem of accessing criminal websites to steal sensitive information can be better solved by various strategies. Based on a comparison of different strategies, Yara's rules seem to work much better.

Key Words: **YARA rules, Malicious website, Phishing URL, Web Crawling**

1. INTRODUCTION

Malicious web pages are those that contain content that can be used by attackers to exploit end-users. This includes web pages containing criminal URLs that steal sensitive information, spam URLs, JavaScript scripts for malware, Adware, and more. Today, it is very difficult to see such weaknesses because of the continuous development of new strategies. Moreover, not all users are aware of the different types of abusive attackers that can benefit from them. Therefore, if there is an accident on a web page, the user is unaware of it, this tool will help him to stay safe despite his lack of website knowledge. Additionally, if the URL itself is marked as a criminal URL, the user will be protected from that website. Python language, an open-source, with the help of a variety of libraries, easy-to-understand syntax, and many resources, proves to be the best way to use a learning machine. One way to do this is to check the URL listed in the so-called malicious websites for a reliable source. But the drawback of this method is that the list is incomplete that is, it grows every day. And, because of such a large list, the downtime of the system will always grow which can be frustrating for the user. Therefore, we use a web crawling method, in which the URL can be classified by Yara rules. It takes the web traffics, web content and Uniform Resource Locator (URL) as input features, based on these features classification of phishing and nonphishing websites are done. Companies have used powerful computers to filter supermarket scanner data and analyze market research reports for years. However, the continuous improvement of computer processing power, disk storage, and mathematical software significantly increase the accuracy of the analysis while reducing costs. The analysis and discovery of useful information from the World Wide Web pose a major challenge for local researchers. Such types of events for gaining valuable information by extracting data mining techniques are known as Web Mining. Web mining is the use of data mining techniques to automatically find and extract information from the web. The goal is to ensure safe browsing regardless of the website the user wishes to visit. Even if a user decides to visit a criminal website to steal sensitive information, steps will be taken to protect the user from harm.

2. Background

Our work relates to research in the areas of heuristic web content detection, machine learning web content detection, and deep learning document classification. One body of web detection work focuses solely on using URL strings to detect malicious web content. [1] proposes a crawling web page for detecting malicious URLs. They focus on using manually defined features to maximize detection accuracy. [2] Also focuses on detecting malicious web content based on URLs, but whereas the first of these uses a manual feature engineering-based approach, the second shows that learning features from raw data with a deep neural network achieves better performance. [3]

Uses URLs as a detection signal, but also incorporates other information, such as URL referrers within web links, to extract hand-crafted features which they provide as an input to both SVM and K-nearest neighbors classifiers. All of these approaches share a common goal with our work, the detection of malicious web content, but because they focus only on URLs and related information, they are unable to take advantage of the malicious semantics within the web content. While URL-based systems have the advantage of being lightweight and can be deployed in contexts where full web content is not available, our work focuses on HTML files because of their richer structure and higher information content. Since these approaches use orthogonal input information, there is certainly room for HTML-based and URL-based approaches to be combined into an even more effective ensemble system. A body of work including [4], [5], [6], and [7] attempts to detect malicious web content by manually extracting features from HTML and JavaScript, and feeding them into either a machine learning or heuristic detection system. [4] Proposes an approach that extracts a wide variety of features from a page's HTML and Javascript static content, and then feeds this information to machine learning algorithms. They try several combinations of features and learning algorithms and compare their relative merits. [5] eschews machine learning and proposes manually-defined heuristics to detect malicious HTML, also based on its static features. [6] Also utilizes a heuristic-based system, but one which leverages both a JavaScript emulator and HTML parser to extract high-quality features. Similarly, [7] proposes a web crawler with an embedded JavaScript engine for JavaScript DE obfuscation and analysis to support the detection of malicious content. The approach we propose here is similar to these efforts in that we focus on a detailed analysis of HTML files, which include HTML, CSS, and embedded JavaScript. Our work differs in that instead of parsing HTML, JavaScript, or CSS explicitly, or emulating JavaScript, we use a parser-free tokenization approach to compute a representation of HTML files. A parser-free representation of web content allows us to make a minimal number of assumptions about the syntax and semantics of malicious and benign documents, thereby allowing our deep learning model maximum flexibility in learning an internal representation of web content. Additionally, this approach minimizes the exposed attack surface and computational cost of complex feature extraction and emulation code. Outside of the web content detection literature, researchers have made wide-ranging contributions in the area of deep learning-based text classification. For example, in a notable work, [8] shows that 1-dimensional convolutional neural networks, using sequences of both unsupervised (word2vec) and fine-tuned word embedding, give good or first-rank performance against a number of standard baselines in the context of sentence classification tasks. [9] Goes beyond this work to show that CNNs that learn representations directly from character inputs perform competitively relative to other document classification approaches on a range of text classification problems. Relatedly, [8] proposes a model that combines word and character-level inputs to perform sentence sentiment classification. Our work relates to these approaches in that, because our model uses a set of dense network layers that apply the same parameters over multiple subdivisions of a file's tokens, it can be interpreted as a convolutional neural network that operates over text.

3. Proposed Methodology

The proposed system is detecting the phishing website using URL scores and textual data analysis. In our proposed system the user can enter the URL in the application then it fetches the website, and then displays the result that the given website is safe or malicious.

The following actions will take place when a user surfs a website:

- The URL of the website will be captured and analyzed, and a final percentage score will be generated based on domain length, URL length, Unique character ratio, brand name presence, and global, global rank of a website.
- After that it will analyze the textual data of that website using a web crawler as classified in YARA rules, and it will detect whether the data is irrelevant or not.
- Based on both results(URL score + textual data analysis), if the data is found irrelevant the system will display the alert message with output.

Following are some YARA rules classified for web crawlers:

- Whether website redirects using JavaScript and HTML, meta refresh tags, mouseover tags.
- Are there any hidden links in the background and the same color link.
- Is the password stored in cookies?
- JavaScript should be embedded in a website.
- Passing parameters through pages that are already authenticated without even knowing using forms.
- Injecting scripts into a website that makes users' credentials at risk.
- when an iframe is there, the website that, checks whether it's at the top and if not, it makes itself on the top.
- techniques to defeat frame busting and using sites in iframe illegally
- is there any key capturing malware website? (Key Logger)

4. Result

In this system, the user can check whether the URL is phishing or not as shown in fig. 1. The user is supposed to enter the URL of the website and click on submit button. After they clicked the submit button, the web crawler will fetch the URL and notify the user whether the site is phishing or not. So that user is secured from the phishing website. Fig.2. If the website is malicious then it will display a probable percentage of malicious websites and it is calculated by using the score of domain length, URL length, unique character ratio, brand name, and global rank of the website, along with textual data analysis. As shown in Fig.3. If the website is safe then it will display a message that is 'Safe Website'.

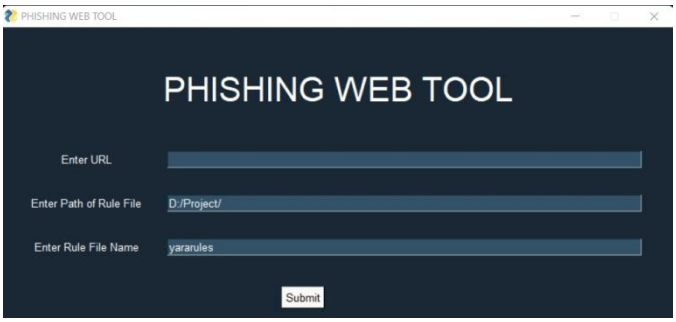


Fig.1. Phishing Web Tool

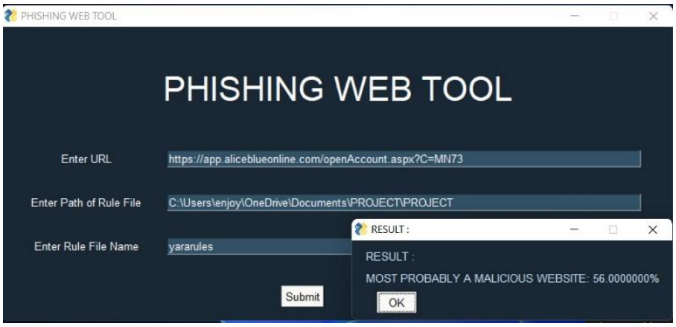


Fig.2. Phishing Website result

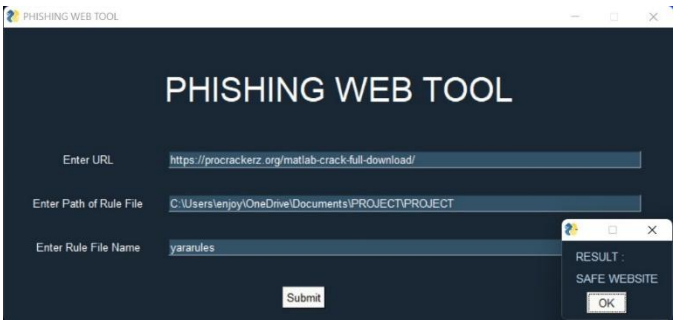
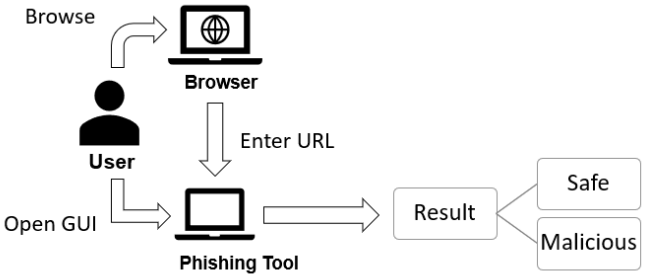


Fig.3. Safe website result

5. System Architecture:



6. CONCLUSIONS

The proposed System aims to implement the detection of phishing websites using web crawling. This task is done by extracting the options of the website via a uniform resource locator once the user visits it. The obtained options can act as taking a look at information for the model. The most task of this method is to sight the phishing website and alert the user beforehand thus on stop the users from obtaining their credentials used. If any user still needs to proceed, it is often done at their own risk.

REFERENCES

[1] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1245–1254, ACM, 2009.

[2] J. Saxe and K. Berlin, "expose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys," arXiv preprint arXiv:1702.08568, 2017.

[3] H. Choi, B. B. Zhu, and H. Lee, "Detecting malicious web links and identifying their attack types.," WebApps, vol. 11, pp. 11–11, 2011.

[4] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in Proceedings of the 20th international conference on World wide web, pp. 197–206, ACM, 2011.

[5] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of malicious web pages with static heuristics," in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian, pp. 91–96, IEEE, 2008.

[6] K. P. Kumar, N. Jaisankar, and N. Mythili, "An efficient technique for detection of suspicious malicious web site," Journal of Advances in Information Technology, vol. 2, no. 4, 2011.

[7] B. Feinstein, D. Peck, and I. SecureWorks, "Caffeine monkey: Automated collection, detection and analysis of malicious javascript," Black Hat USA, vol. 2007, 2007.

[8] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts.," in COLING, pp. 69–78, 2014.

[9] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, pp. 649–657, 2015.

[10] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on, pp. 11–20, IEEE, 2015.

[11] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[12] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.