EVENT-DRIVEN MALICIOUS URL EXTRACTOR

***Abstract*— Cyber-attacks are attacks that are commonly carried out in order to obtain sensitive information or disrupt internet-based services. Recent occurrences, both internationally and locally, have shown an influx of these attacks expanding rapidly through the use of malicious URLs (Uniform Resource Locators). Traditional measures, including such blacklisting malicious URLs, make it extremely difficult to respond to such attacks in a timely and efficient manner. Most existing solutions remain restricted in terms of scalability and proactive user safeguarding in situations when freshly formed URLs are correlated with a recent event, such as Covid-19 related frauds. The proposed solution is presented with the primary aim of addressing traditional system limitations and offering an interface for users to protect themselves by detecting phishing/malicious URLs in real time. In this research, we will examine extracting user-input event-related keywords and leveraging NLP (Natural Language Processing) algorithms to match them with the accompanying URL (Uniform Resource Locator) token data to determine whether the URLs are malicious or benign.**

Keywords— Malicious URL Detection, Machine Learning

I. INTRODUCTION

A. Background on malicious links

The importance of the World Wide Web has been increasing rapidly. Ironically, these advancements are associated with new techniques for attackers to attack and mislead clients. There are numerous techniques for executing such attacks, including drive-by downloads, social engineering, phishing, and many others. Due to the astounding development of new security risks, rapid changes in new IT (Information Technology) resources, and a scarcity of security experts, the limitations of conventional security approaches are becoming increasingly critical. The purpose of this research is to discover ways of preventing such attacks. The majority of these attack tactics are identified by disseminating forged URLs (or the spreading of such URLs forms.)

These malicious attacks can often result in large-scale cyber-attacks or disruptions. Moreover, cybercriminals have been using a multitude of approaches to make malicious websites resemble as legitimate as possible since the beginning. As with incidents like Covid – 19, attackers intend to take advantage of this event-related data to deceive users into landing on malicious phishing pages in order to steal sensitive information. One popular method is to make the domain name include terms which are like a specific situation, and the visitor would want to visit the website because of this affliction. A case in point would be the COVID-19 phishing URLs, which is using keywords as covid-19, WHO (World Health Organization), or vaccines to mislead consumers into accessing the phishing pages. The use of the keywords "covid" and "corona" in certificates and URLs surged to 14,940 in March, according to F5 Labs' report. [1]

B. Available solutions

Much previous research has been using both machine learning and non-machine learning approaches to automate the identification of phishing URLs. The preponderance of these techniques offers consumers the option of blacklisting URLs or signatures. Traditional measures such as blacklisting are not feasible as it is not practical for average consumers to maintain such vast databases of malicious URLs. To resolve these concerns, researchers have been focusing on using machine learning algorithms to determine whether such a URL is malicious.

One of these studies examines overall hostname length, URL length, and tokens in URLs using lexical analysis of URLs. The authors speculate that additional characters are frequently inserted within phishing URLs to make them appear authentic. Using alphabet entropy, this can be used to detect randomly generated malicious URLs. After features extraction from the URLs list, the data is then processed by a classifier, which will classify the data into malicious and non-malicious classes depending on lexical features. [2]

Another research adopts the “Monarch” framework, which proposes a real-time web crawling technique to assess whether URLs are malicious or benign. “Monarch” is said to be capable of processing up to 15 million URLs each day. The approach works on the same principle as before, collecting features from URLs and feeding the data into a classifier to construct a model for detecting malicious URLs. What's novel is the improved memory efficiency and algorithmic updates, which include a linear classifier based on logistic regression and the L1-regularizer to produce sparse models. [3]

Paper [4] propose a solution for increasing the scalability of the keyword matching with URL tokens when larger scale of data is at analysis phase. This solution, “online learning” is much more efficient than the traditional batch processing

methods. Authors categorize the online learning algorithms into two main categories depending on their application at malicious URL detection: (i) First-order online algorithms, and (ii) Second-order online algorithms.

II. Proposed Methodology

The proposed Event-Driven Phishing URL Extractor has the capability of,

- Populating keywords related to the user input keywords.
- Generating keywords/ tokens from URLs.
- Identifying and classification of matching URLs with keywords for specific events.
- Determining the matched URL is malicious or benign.

study focuses on the identification of related URLs from user inputted keywords.

To generate efficient matching keywords, imported words will be loaded into data frames, split, and filtered to remove retweets and punctuation, among other characteristics.

B. Comparison Engine

Initially, both malicious and benign URLs are ingested into the comparison engine, which utilizes 'Regexp' tokenization to query for event-related keywords and obfuscated variants. Before matching the keywords obtained by the populator module, these tokens stem to remove extraneous sections of the URLs. For example, variants of the word images will be stemmed into image.

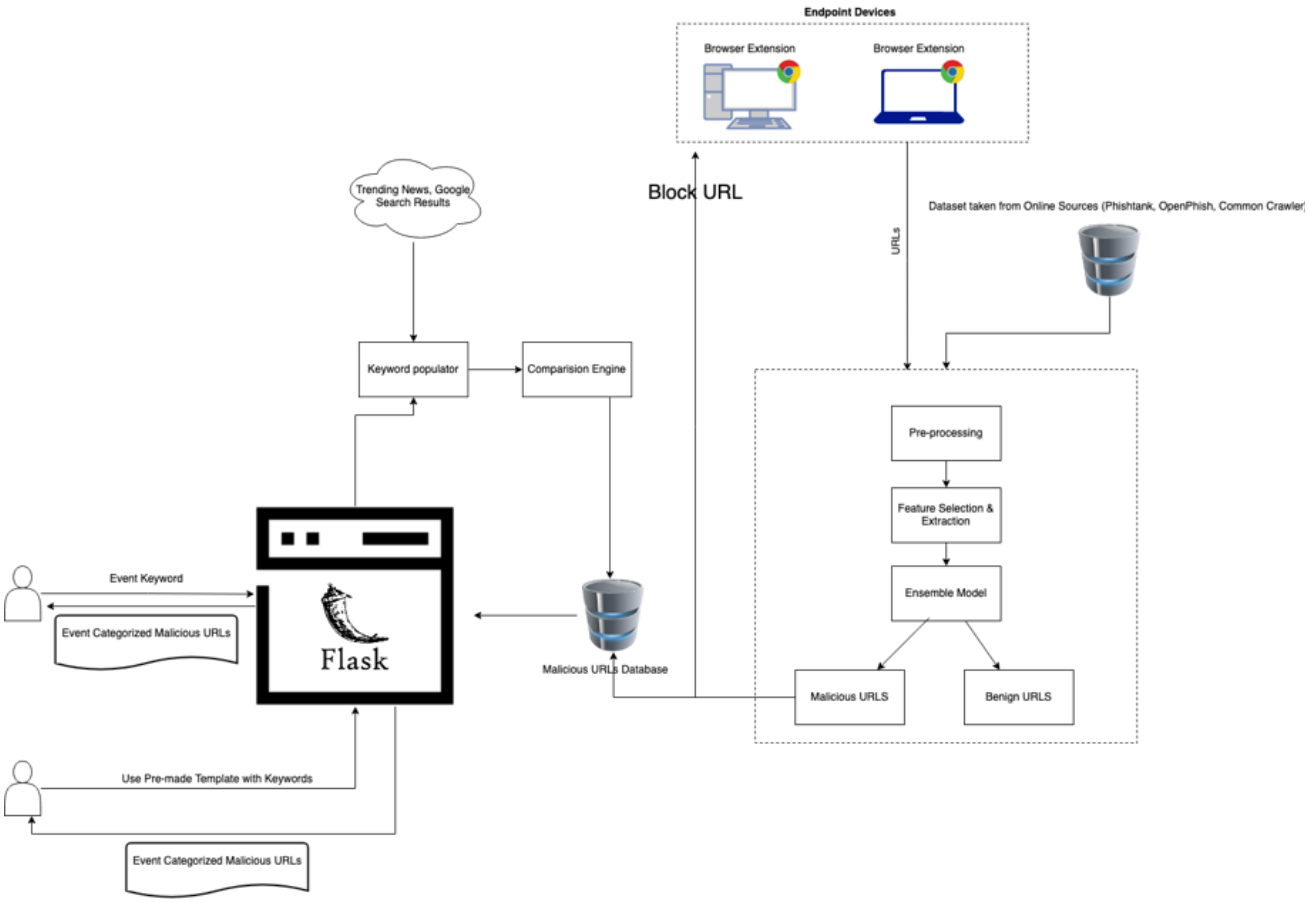

Fig. 1. Malicious URL Extractor Architecture

A. Keyword populator

To combat influxes of malicious URLs correlated with a specific event, we propose a model that detects malicious URLs correlated with keywords associated with such events. Keyword populator will ingest the user input keywords via the web interface built using the Python Flask framework, and search and retrieve related keywords from tweets in real time leveraging the Twitter API (Application Programming Interface) to match and populate inputted keywords, and the

C. Feature Reduction

Feature Reduction is a main objective which is followed along with the other main objectives to achieve the product and the sub objectives that comes with this is Feature Classification, Feature Extraction, Feature Analysis and Feature Reduction which is done in the end. Feature Reduction or Dimensionality Reduction as it is known is used to reduce a dataset with high dimensionality or features as we call it, to lower and manageable dimensionality so that we can easily use that dataset to work on a model. Having less

features or less dimensionality helps to ensure the efficiency of an algorithm, higher speed of a system and ensures less training time/computation time. Having a set of features which are small and having the needed basic characteristics of the input values in a system helps to ensure efficient and effective functionality of a system where we need to work with high amount of data, and therefore Feature Reduction is important. [5]

Feature Classification is done to carry out the processes of feature extraction, analysis, and reduction. We must first understand the parts of a URL and then we can move on to understand how they can be altered by attackers to look legitimate enough to trick users into navigating into sites with these malicious URLs. The protocol name used to access the web page is the first part of a URL, the organization name in the hosting server is represented by the subdomain and the second level domain name (SLD), and then we can find the top-level domain which shows the domains which could be found in the DNS's root zone. [6]

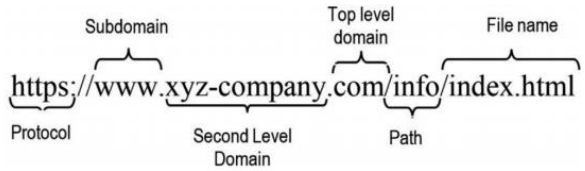

Fig. 2. Components of a URL [6]

The domain name is the unique and most critical part of the URL which is made up of the TLD (Top Level Domain) and SLD. Attackers use these domain names for their attacks by further adding random characters or words or special symbols to make the attack successful. [6]

Feature classification will be achieved by trying to classify the features found out from malicious URLs and the highly specific features which are independent of each other are used for the classification process. A total of 24 features could be found in recent research done in the year 2021. It shows that using these features would be effective so to break it down further to a much smaller number of features, first we will take these features into consideration. [7]

S. No.	*Feature*	*Explanation*
1	*URL Length*	*Length of the URL string*
2	*Max Word Length*	*Maximum length of a word present in URL*
3	*Domain Length*	*Total domain length*
4	*No of dots*	*Total Number of dots present in URL string*
5	*No. of ?*	*Total Number of question marks present in URL*
6	*No. of /*	*Number of times forward slash occurred in URL string*
7	*Numeric count*	*Total numerical letters present in URL string*
8	*Special characters*	*Total Count of special Characters such as @, !, #. etc.*
9	*.exe or .install*	*Use of .exe or .install in URL*
10	*.com*	*Presence of .com in URL*
11	*.php*	*Presence of .php in URL*

Fig. 3. Classified URL Features-1 [7]

S. No.	*Feature*	*Explanation*
12	*.gov*	*Presence of .gov in URL*
13	*.org*	*Presence of .org in URL*
14	*.edu*	*Presence of .edu in URL*
15	*https*	*Presence of https in URL*
16	*.htm or .html*	*Presence of .htm or .html in URL*
17	*.net*	*Presence of .net in URL*
18	*.info*	*Presence of .info in URL*
19	*.js*	*Presence of .js in URL*
20	*Tempting words*	*Total Number of temptation words like money, free etc.*
21	*Offensive words*	*Total Number of offensive words present in URL*
22	*Image*	*Presence of image word in URL*
23	*Login or upload*	*Presence of Login or upload word in URL*
24	*IP address*	*Directly presence of IP address as domain*

Fig. 4. Classified URL Features-2 [7]

The above proposed list of features from the research is said to have features from both genuine URLs and malicious URLs as well. For example, genuine URLs usually contain top level domain names such as given in features 12, 13 and 14 which are *.gov*, *.org* and *.edu* but malicious URLs usually contain extensions such as *.htm / .html*, *.info* and *.js* given in feature numbers 16, 18 and 19. [7]

Further breaking down of feature classification brings us to 3 primary features that will break into sub features once again which will now be related to only malicious URLs. The 3 key features are address bar-based features, domain-based features, and html & JavaScript based features. The following table shows the 3 key features with their sub features in the respective columns.

Address Bar based features	**Domain based features**	**HTML & JavaScript based features**
Domain of URL	DNS (Domain Name System) Record	iFrame Redirection
IP (Internet Protocol) Address in URL	Website Traffic	Status Bar Customization
Presence of @	Age of Domain	Disabling Right Click
Length of URL	End Period of Domain	Website Forwarding
Depth of URL		
Redirection // in URL		
http/https in Domain name		
Use of URL shortening services		
Prefix/Suffix "- "in Domain		

Feature extraction is the next process as we go further through the process of feature reduction and using python-

based coding on building up the program for feature reduction using datasets from the sites Phishtank (malicious URLs) and UNB (legitimate URLs) is done in order to achieve the goal of feature reduction by analyzing the features extracted from the datasets. Each and every feature of the URL is set to be tested in a way to produce a Boolean output of 1 for malicious URLs or 0 for legitimate URLs. The python code that was written checks for the previously mentioned features in both the datasets that are downloaded from the said websites and then those URLs would be extracted. URLs would be chosen randomly from both the datasets and then they would be checked for the features. Next, number of URLs are collected randomly from each of the datasets. Then the features of legitimate URLs will be stored in a data frame. Then the same process is carried out with malicious URLs in the same manner. Then the concatenated data table is stored in a .csv file.

Feature analysis is carried out as the next process. The previous dataset which was created in feature extraction will be used for this process. It will be loaded into a data-frame. Then it will be checked for NaN values and then rows will be removed. NaN stands for Not a Number and is a numeric data type used to represent any value that is undefined or unpresentable. For example, 0/0 is undefined as a real number and is, therefore, represented by NaN. Then we will get a total of how features were listed for the URLs to see which features are the most common features.

D. Ensemble Model

Ensemble modeling is a process in which several different models are formed to build predictive models, either using a range of modeling algorithms or different training data sets.

Extending the recommendation [8] of using Bi-directional LSTM with CNN, the accuracy and false-positive rates tend to be lower. Our test environment used a LSTM/CNN ensemble against a Bi-Directional LSTM/CNN ensemble to evaluate the performances. A 100-epoch cycle was used a common parameter for both ensemble which produced 92% and 95% accuracy respectively.

E. Federated Learning for malicious URL detection

Federated Learning is a concept for decentralizing training used in Machine Learning. This collaborative learning approach contrasts the traditional centralized machine learning techniques. It interactively learns a shared model without transferring the training data to a central location.

In terms of cybersecurity domain, detecting malicious URLs is a well-researched problem. There are existing solutions for this problem relying on centralized learning which uses several techniques ranging from pattern mining based [9], deep learning-based approaches [10] to semantic information extraction from URLs [11].

In practice, there are rule based systems that combines machine learning approaches to blacklisting URLs. Ensuring the privacy and securing the copious amounts of data that is being generated is one of the challenges we have encountered in our research when distinguishing malicious URLs with benign URLs.

To overcome this challenge, we will be using Federated Learning. Thus, the collected data are examined with the help of Federated Learning minimize the problem associated with data collecting when distinguishing the URL component.

III. Deploying the Malicious URL Extractor

A. Web Interface

The user is the primary focus of the proposed solution, which would be delivered through an intuitive web interface built with Python Flask framework. Users can utilize the user interface to input keywords, generate visualized data which correlates to results, or export event-based URL databases in.CSV format.

B. Endpoint Browser Extension

Our solution offers a browser extension which endpoint device users can install to harvest URLs in JSON (JavaScript Object Notation) format utilizing JavaScript with explicit permission, in order to achieve an efficient user experience. The classifier model will use these URLs to evaluate whether they are malicious or benign. Moreover, to make the project more user-friendly, the browser extension will automate the blacklisting of identified malicious URLs from the classifier. Understanding the privacy concern of the users, our proposed solution incorporated Federated Learning which contrasts the traditional centralized machine learning techniques, which is based on distributed datasets without sharing raw data while preserving the data privacy. [12]

IV. Conclusion

This paper presents the design of the Event- Driven Malicious URL Extractor utilizing an efficient ensemble model and provide a user-friendly secured application for users to safeguard themselves from malicious content. The authors extended the current research done in this domain to assemble an efficient system and adding privacy-related features such as Federated Learning.

The authors believe that the following features could be further enhanced in the system.

1. Improved scalability of the overall system
2. User experience of the website
3. Increasing the number of epochs for more rigorous testing of the ensemble model

V. References

[1] F. L. D. Warburton, "2020 Phishing and Fraud Report," [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/2020-phishing-and-fraud-report. [Accessed 21 02 2021].

[2] I. M. M. A. R. A. H. L. N. S. a. A. A. G. Mohammad Saiful, "Detecting Malicious URLs Using Lexical," in *University of New Brunswick, Fredericton, NB, Canada*, Fredericton, 2016.

[3] C. G. J. M. V. P. a. D. S. Kurt Thomas, "Design and evaluation of a real-time url spam filtering service," in *IEEE Symposium on. IEEE*, 2011.

[4] C. L. S. C. H. DOYEN SAHOO, "Malicious URL Detection using Machine Learning: A Survey," 2019.

[5] P. Sharma, "The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)," Analytics Vidhya, 27 August 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/.

[6] E. B. O. D. B. D. Ozgur Koray Sahingoz, "Machine learning based phishing detection from URLs," *Elsevier*, p. 14, 2018.

[7] B. S. Yogendra Kumar, "A lightweight machine learning based security framework for detecting phishing attacks," *COMSNETS*, p. 5, 2021.

[8] B. J. a. S. R. V. Yazhmozhi, "Anti-phishing System using LSTM and CNN," EEE International Conference for Innovation in Technology (INOCON), Bengaluru, 2020.

[9] D. Huang, K. Xu and J. Pei., "Malicious URL detection by dynamically mining patterns without pre-defined elements.," *World Wide Web 17,*, p. 1375–1394, 2014.

[10] V. R, S. S and S. K. a. M. Alazab, "Malicious URL Detection using Deep Learning," January 2020.

[11] S. Marchal, K. Saari and N. S. a. N. Asokan, "Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, Japan, 2016.

[12] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang and J. Liu, "Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[13] D. Warburton, "2020 Phishing and Fraud Report," F5 Labs, [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/2020-phishing-and-fraud-report. [Accessed 21 02 2021].