Phishing Website Detection Using Machine Learning

Prateek Gupta and Archana Singh

Abstract Phishing costs around billions of dollars per year to the Internet users. Novel phishing techniques for instance spoofing in between trusted websites on the Internet are leveraged to phish target's account information, login credentials and personally identifiable information such as email Id, date of birth, biometrics and passwords. Most commonly attackers use phishing software and spam emails for stealing personal information to collect financial accounts details like credit card details and credentials. In the recent study, it was identified that phishing attacks account for more than 80% of reported security incidents and 94% of them are via email. The research work presented here is on a multitude of strategies which are used for detection of malicious and phishing sites depending on their various lexical features. Information gathered from the study of malicious and phishing sites is then used for lexical features assessment, and further, to analyze and to improve upon the algorithm used for the detection of malicious and phishing sites. Most organizations today use rule-based engines for phishing detection that do not proactively scale for phishing attacks without additional rules deployment. From the recent study, one could gather that for an organization, an improvement in phishing detection does have a positive impact on net revenue. An associated data point here is that $17,700 is lost every minute due to phishing attacks, thus a need for a comprehensive solution to phishing attacks. The paper provides an analysis of various methods used for detecting phishing websites by using machine learning and classification techniques based on lexical features. Machine learning-based techniques leverage natural language processing and other classification techniques like logistic regression, support vector machines and random forest [1, 2]. In order to have a comprehensive machine learning-based solution, training data is required to possess lots of relevant and non-correlated features. A comprehensive learning algorithm can effectively determine not previously classified URLs with a better accuracy. Here,

P. Gupta (✉)
Adobe, Ghaziabad, India
e-mail: pratgup@adobe.com

A. Singh
ASET, Amity University, Noida, UP, India
e-mail: asingh27@amity.edu

S. Fong et al. (eds.), *ICT Analysis and Applications*, Lecture Notes in Networks and Systems 154, https://doi.org/10.1007/978-981-15-8354-4_19

our purpose is to study various machine learning algorithms that can be leveraged for safeguarding users from spoofed websites and help them not fall in the trap of phishing by detecting these websites early. Previous work done on the subject is also studied and compared against for accuracy.

Keywords Logistic regression · Random forest · Support vector machine · Security · PII · Phishing · Malicious

1 Introduction

The conventional method used to detect malicious websites is based on a predefined dictionary of blacklisted websites. A website gets blacklisted based on user's feedback who encountered a malicious intent. However, the disadvantage of the blacklisting method is that the websites which are not found previously as malicious cannot be predicted whether they are malicious or not. Blacklisting is a static method of detecting URLs malicious nature. Blacklisting method is mostly accurate and reliable. Nonetheless, it cannot be the only way of detecting malicious sites as the blacklisting method would not work accurately in today's time of dynamic malicious URLs. Dynamic here refers to auto-generating malicious URLs. As can be seen from the latest APWG phishing activity trends report, the number of phishing websites is significantly increasing year on year. Magnitude of increase can further be envisaged by the fact that there are many which get detected, blacklisted and closed—despite which graph is having an upward trend (Fig. 1).

Another disadvantage of the blacklisting method is that it is highly dependent on the incidents reported by the users. The heuristic classification is an improvement of the previously mentioned blacklisting approach. In this approach, the signature of the previously existing malicious URL and the signature of the new URL are matched. Even though a heuristic classification method like the blacklisting method is also highly effective, it cannot cope up with today's evolving phishing attack techniques [3]. Another demerit to consider is that both heuristic and blacklisting methods become more and more complex as the database of signatures, and blacklist

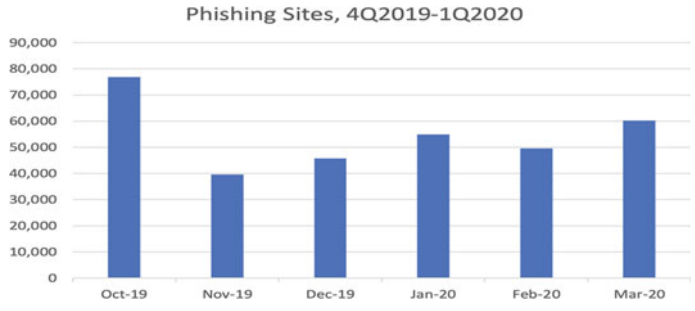

Fig. 1 Phishing sites from quarter year 2019–2020 (*source* Internet)

Fig. 2 Most targeted phishing sectors 2019–2020 (*source* Internet)

websites keep on increasing exponentially daily and at times even hourly. Phishing is not restricted to a sector and is well distributed. Each sector has its own set of phishing experts who deploy unique methods and semantics-based attack structure, which is designed specifically for that domain and mainly exploits the computer users' vulnerabilities (Fig. 2).

2 Features of URLs

2.1 Blacklist Features

As we have already discussed in the previous sections that the conventional method used to detect malicious websites is to make a list of blacklisted malicious websites. These blacklists are built using various methods such as feedback taken from humans whenever they encounter these websites, and the human feedbacks are very accurate in nature as they are verified by the humans itself. But the disadvantage of this blacklisting method is that the websites which are not found previously as malicious cannot be predicted whether they are malicious or not. As mentioned above, this

method is highly dependent on the incidents reported by the users. But it plays a vital role while the training of a machine learning algorithm, and hence, is considered in the feature list.

2.2 Lexical Features

The lexical features are also known as URL-based features. Foremostly, URL of the website is analyzed in order to detect malicious websites. In this length of the URL is taken into consideration, the number of digits in a URL is counted and typo squatted URLs are scanned (e.g. www.goggle.com). In addition, the number of sub-domains in the URL and whether the top-level domain is commonly used or not is also examined. Further, the algorithm is advanced to consider the number of dots in a URL and to dynamically identify domain and sub-domains.

Lexical features also examine tokens in hostname. To list a few: '?', '+', '%', '=', '.', etc. These features are helpful in verifying behavior of a webpage. For instance, multitude of tokens attributed to slashes may indicate denial of service attacks. The domain name can directly indicate a malicious website which has been previously blacklisted.

2.3 Features: Examine IP Address of the URL in Address Bar

For this feature, let us leverage the IP address of the given domain as alias, for example: "http://128.68.1.94/fakewebpage.html". To analyze cases wherein end users are sure that a person or a bot is seeking and trying to perform data theft on their personal information, IP address is sometimes even converted into the hexadecimal formatted code, as shown here: "http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html".

$$\underline{Rule} : \text{IF} \begin{cases} \text{If The Domain Part has an IP Address} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Lengthy URL conceals the malicious element
Usually, what phishers do is, they make use of lengthy URLs, to make sure that the suspicious part in the URL is not visible. To strengthen our work and study, we figured out the size of URL in the given tuples of our dataset and tried to harness a basic URL length. The outcome and result reflected that phishing URLs mostly have the length of the URL similar to or greater than 54 alphabets and characters. Outcome was strengthened by studying the dataset we used wherein we figured that suspicious URL sizes are actually greater than 54 alphabets.

$$\text{Rule: IF} \begin{cases} \text{URL length} < 54 \rightarrow \text{feature} = \text{Legitimate} \\ \text{else if URL length} \geq 54 \text{ and } \leq 75 \rightarrow \text{feature} = \text{Suspicious} \\ \text{otherwise} \rightarrow \text{feature} = \text{Phishing} \end{cases}$$

We were in the position to be able to reconsider and update this particular attribute or feature, with the help of using a technique, which is totally based upon the frequency, and this helped us in increasing the relative accuracy of the algorithm.

Very Short URLS or "TinyURL" Using URL trimming services
Trimming services used by many phishers shorten the URL. It is a very smart method on the Internet, in which a URL which is relatively smaller in size but can cause equally or more detrimental attacks. This can be obtained and done, with the help of "HTTP. Redirect" upon a URL, which is small, that can be used to link toward the website, which has a very lengthy URL name. For example:

$$\underline{Rule} : \text{IF} \begin{cases} \text{TinyURL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

URL's having "@" Symbol
When a phisher uses the "@" character in the URL, it results in the leading of the specific web browser to ignore values preceding the "@" symbol.

$$\text{Rule: IF} \begin{cases} \text{Url Having@Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Redirecting using "//"
Existence of character "//" in the given URL path implies that the end user on click of URL will be sent to a non-identical webpage on the Internet. An example of such a website URL is "http://www.legitimate.com// http://www.phishing.com" [3, 4].

During our research, we examined a similar URL where there is a presence of "//". We identified that, if a URL starts along "HTTP", this implies that "//" will mostly appear in the sixth position of the URL. However, in a scenario wherein Domain URL has "HTTPS", then "//" will most likely be at position 7.

$$\text{Rule: IF} \begin{cases} \text{The Position of the Last Occurrence of "/" in the URL} > 7 \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Using (−) to the domain
This hyphen of dashed symbol is very rare in use in the good URLs. Attackers tend to attach prefixes and suffixes, which are separated by using the symbol of (−) in the domain name, in result of that, the end users tend to feel that he/she is handling the legitimate URL. For example:

$$\text{Rule: IF} \begin{cases} \text{Domain Name Part Includes}(-)\text{Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Sub-domain and Multi-sub-domains
If in case there is a situation, in which the number of given dots in the URL is much larger in terms of the number than 2 or 3, then we will identify such URL as "suspicious", since it has only one and not more than one sub-domain.

$$\text{Rule: IF} \begin{cases} \text{Dots In Domain Part} = 1 \rightarrow \text{Legitimate} \\ \text{Dots In Domain Part} = 2 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

HTTPS (Hypertext transfer Protocol secured using Socket layer/TLS)
HTTPS in URL is crucial and helps in giving the impression of the webpage accuracy and reliability, but this technique cannot be used in isolation as it is not adequate enough. The writers of different research papers have given suggestions like to verify the assigned legitimate certification of the HTTPS which is to verify with the issuers if the domain is certified and the age of certification is valid. Certification authorities can be consistently listed among out most of the top renowned names like "geotrust, GoDaddy…, etc.". On further research and testing the dataset, we were able to find out that the minimum age of the given URL is of two and more years [5, 6].

$$\text{Rule: IF} \begin{cases} \text{Use https and Issuer Is Trusted and Age of Certificate} \\ \quad \geq 1 \text{ Years} \rightarrow \text{Legitimate} \\ \text{Using https and Issuer Is Not Trusted} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

Domain Registration Length
On the basis of the various past studies, it can be concluded that the phishing webpage lives only for a short period of time, while on the other hand legitimate domain names are generally paid for a longer duration, say for multiple years. During the analysis of the dataset in consideration, it was identified that the big and longest fraud domains are most likely to be used for not more than a year time frame.

$$\text{Rule: IF} \begin{cases} \text{Domains Expires on} \leq 1 \text{ years} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Favicon
Favicon is a picture associated with the specific website. Many users, which exist on the Internet, acting as an agent, such as graphic-oriented browsers and news apps,

leverage favicon to optically remind of the webpage and to add falcon as URL's identity in the address bar. In scenarios wherein the favicon is reloaded on refresh of URL and it apparently is different to the one shown in the address bar, then in that case, the website is considered as a malicious and a phishing website.

$$\text{Rule: IF} \begin{cases} \text{Favicon Loaded From External Domain} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Using Non-Standard Port

Non-standard port feature is extremely beneficial in validating if a service is from the authentic server. To prevent organization from phishing attacks, it is good to only open and close the ports as per requirement and keep them always up to date. Multiple security firewalls, proxy and NAT servers are placed to protect confidential data and to block the susceptible ports [7, 8].

$$\text{Rule: IF} \begin{cases} \text{Port \# is of the Preffered Status} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Using "HTTPS" Token in the domain element of the URL

The attackers can try to add the "HTTPS" token in the domain element of the URL, in order to disorient the users. For example:

$$\text{Rule: IF} \begin{cases} \text{Using HTTP Token in Domain Part of The URL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Abnormal-based Features Request URL

It is validated if the different aspects in a webpage like image, visuals, graphics, videos and sounds are being directed to another domain. In a legitimate webpages, the webpages, and the content of webpage are mostly unique and have correlation only with other webpages in the same domain [5, 9, 10].

$$\text{Rule: IF} \begin{cases} \%\text{ of Request URL} < 22\% \rightarrow \text{ Legitimate} \\ \%\text{ of Request URL} \geq 22\% \text{ and } 61\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{feature} = \text{Phishing} \end{cases}$$

URL of Anchor

An anchor URL is defined using the <a> tag. It is also known as link label and is examined to identify the rankings that the webpage will receive from globally renowned search engines. This feature is used as a "Request URL". Moreover, in this attribute, we validate:

Whether the <a> tag is partially or entirely associated with webpages and if all are corresponding to the same domain and have a similar ranking across search engines.

$$\underline{Rule} : \text{IF} \begin{cases} \%\text{ of URL Of Anchor} < 31\% \rightarrow \text{Legitimate} \\ \%\text{ of URL Of Anchor} \geq 31\% \text{ And } \leq 67\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

Linked webpages are placed with <Meta>, <Script> and <Link> tags During the course of research, multiple methods that can be used in a webpage source code were covered. It was observed that it is acceptable for a website to make use of <Meta>, <Script> and <Link> tags to extract other resources.

$$\text{IF} \begin{cases} \%\text{ of Links in ``<Meta>'', ``<Script>'' and ``<Link>''} < 17\% \rightarrow \text{Legitimate} \\ \%\text{ of Links in ``<Meta>'', ``<Script>'' and ``<Link>''} \geq 17\% \text{ And } \leq 81\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

Server from Handler (SFH) SFHs that contain an empty string and a "about.blank" are considered as suspicious. In addition, if a webpage's domain name is in the SFHs and is non-identical to the parent domain name, it implies a suspicious website.

$$\text{Rule: IF} \begin{cases} \text{SFH is ``about:blank'' Or Is Empty} \rightarrow \text{Phishing} \\ \text{SFH Refers To A Different Domain} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Submitting information to Email Internet forms, which are used to fill the information, require a user to provide his personally identifiable information. These Internet forms are directed to a backed computer known as a server for further dealing and processing. An attacker may redirect this data and information to his own computer and his workspace and his storage. In the subsequent steps, a script running on the server side of the connection can be accessed for his personal use.

$$\text{Rule: IF} \begin{cases} \text{Using ``mail()'' or ``mailto:'' Function to Submit User Information} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Abnormal URL

This kind of attribute can be taken out of the WHOIS database, available as a library in Python.

$$\text{Rule: IF} \begin{cases} \text{The Host Name Is Not Included In URL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

3 Conclusion and Future Scope

As we can see in this paper, malicious URLs are a big problem in today's world. There has been prior research done in this field, and we have added our findings to the same. However, our approach was to see all aspects related to this field. From our research, we understood that many attempts have been made for the detection and prevention of malicious and phishing URLs present on the web. On reading different research papers and publications, we saw that machine learning is the common approach, which is being ardently followed by people who are working in this area. As we know machine learning is vast in its reach and the impact that it can create, so usage of machine learning for detection and prevention from malicious URLs is good and beneficial. As we identified, a URL is made up of different parts, URLs are just not like a name of a person and have many of their own characteristics. URLs are made up of different elements like its domain, sub-domain, port address, etc. Therefore, what machine learning models do is that it learns tons of URLs and tries to find the similarity between them. So, URLs having the same type of structure are classified as a particular class. Therefore, if we give our model the data of malicious URLs, it will go through every one of them. Then it will try to find similarity between them and classify them as a class of URLs. So when we give any URL which is of the same structure as malicious URLs, a machine learning model will identify that this is a phishing or a malicious URL. So, in machine learning, different algorithms and techniques are present to train a classification model, and each algorithm has its own benefits and disadvantages. Each of them has a different way of learning.

In our future work, the work can be extended by considering more features of URL phishing, and analysis can be done using deep learning techniques and transfer learning. The work can be extended for the dark web to provide more secure solutions.

References

1. Sahingoz, O. K., et al. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications, 117,* 345–357.
2. Jain, A. K., & Gupta, B. B. (2018). PHISH-SAFE: URL features-based phishing detection system using machine learning. In M. Bokhari, N. Agrawal, & D. Saini (Eds.), *Cyber Security. Advances in Intelligent Systems and Computing* (Vol. 729). Singapore: Springer.
3. Buber, E., Dırı, B., & Sahingoz, O. K. (2017). Detecting phishing attacks from URL by using NLP techniques. In *2017 International Conference on Computer Science and Engineering (UBMK)*, Antalya (pp. 337–342). https://doi.org/10.1109/ubmk.2017.8093406.
4. APWG. (2020, December). Phishing activity trends report, 1st Quarter 2020. Technical Report.
5. Vargas, J., Correa Bahnsen, A., Villegas, S., & Ingevaldson, D. (2016). Knowing your enemies: Leveraging data analysis to expose phishing patterns against a major US financial institution. In *2016 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 52–61).
6. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2011, May). Learning to detect malicious urls. *ACM Transactions on Intelligent System Technology, 2*(3), 30:1–30:24; Marchal, S., Saari, K., Singh, N., & Asokan, N. (2016). Know your phish: Novel techniques for detecting phishing sites and their targets. In *International Conference on Distributed Computing Systems* (pp. 323–333).

7. Woodbridge, J., Anderson, H. S., Ahuja, A., & Grant, D. (2016, November). Predicting domain generation algorithms with long short-term memory networks. http://arxiv.org/abs/1611.00791.
8. Zhang, J., Porras, P., & Ullrich, J. (2008). Highly predictive blacklisting. In *17th USENIX Security Symposium* (pp. 107–122).
9. Roopak, S., & Thomas, T. (2014). A novel phishing page detection mechanism using html source code comparison and cosine similarity. In *2014 Fourth International Conference on Advances in Computing and Communications* (pp. 167–170).
10. Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In *SIGCHI Conference on Human Factors in Computing Systems* (pp. 581–590).