Prevention From Shorten URL

N. Logeshwaran
Dept of cyber forensics and information security
Dr. MGR Research Institute

Abstract- *A Short Uniform Resource Locator is a compressed form of long web URLs. The Short URLs is easier to remember and use instead of long URLs. A mechanisms for short URL is used in many situations, including posting messages that must accommodate character limits, such as Twitter or SMS and Storing, reading, copying, or listing numerous short URL and engaging potential customers of products and services and engaging users for fun or pranks. The tabu search mechanism is responsible for the selection of assets and the gradient descent search tries to find the optimal weights by minimizing the objective function.*

It may proposed for it make links more manageable, track and compile click data, transformed into social media services, provide users useful features and promote sharing etc. Traditionally, this detection is done mostly through the usage of blacklists.. However, blacklists cannot be exhaustive, and lack the ability to detect newly generated malicious URLs.

Keywords- prevent from url shortner, malicious url identifier, url shortner identifier, protect from url shortner.

I. INTRODUCTION

SCOPE OF PROJECT

URL shortening is the translation of a long Uniform Resource Locator (URL) into an abbreviated alternative that redirects to the longer URL. The original URL shortening service was TinyURL, which was launched in 2002 by Kevin Gilbertson to make links on his unicyclist site easier to share. TinyURL remains popular today; other commonly used URL shorteners include bitly , goo.gl (Google) and and x.co (GoDaddy). Short URLs are preferable for a number of reasons. Long URLs in text can make the accompanying message difficult to read and links can break if they fail to wrap properly. Although most email clients can now correctly handle long URLs, the use and popularity of shortening URLs has increased because of mobile messaging and social media websites, especially Twitter which has a 140-character constraint.

Although URL services often provide users with handy features such as the ability to customize short URLs and track traffic, some security analysts warn that the use of third party services is simply the addition of another attack vector. Many services are free and offer no service level agreement, which means the user must trust the services ability to keep its servers secure. Additionally, shortened links offer the user no clue as to where they lead and can be used to redirect users to infected content. To compensate, some services allow the user to add a special character at the end of the shortened URL.

The addition of the special character allows the person to hover over the link and preview the page it is pointing to.

Reliability and availability are two more concerns. Even if a service guarantees 99 percent uptime, there will still be 3.5 days per year when its shortened links won't work. And as some users have found to their dismay, shortened links may no longer work if the service goes out of business. Short URLs are widely used in specialized communities and services such as Twitter, as well as in several Online Social Networks and Instant Messaging (IM) systems. A study of URL shortening services will provide insight into the interests of such communities as well as a better understanding of their characteristics compared to the broader web browsing community. Short URLs Services From the beginning of the short URL services the use of short URLs had become a norm in SNSs where generally character limitation exists (Twitter has 140 character limit).

URL shortening is a technique on the World Wide Web in which a Uniform Resource Locator (URL) may be made substantially and still direct to the required page. This is achieved by using a redirect which links to the web page that has a long URL. Other uses of URL shortening are to a link, track clicks, or disguise the underlying address. Although disguising of the underlying address may be desired for legitimate business or personal reasons, it is open to abuse .

For example, the long URL https://pypi.python.org/pypi/pythongraph is given to the any short URLs services as bit.ly it returns the short URLs as https://bit.ly/xxxxx. Though short URL services resulted in space, reducing methodology in SNSs but it has resulted in a security breach like cybercrime. The resulted short URLs may be malicious or benign. The malicious short URLs are obfuscate in nature and cannot be identified by traditional methods (blacklisting). The multiple redirection of short URL has made it very difficult to identify the real malicious URLs.

The Benefits of URL Filtering. URL shortening provides a way to block access to websites. It can also be used to secure sites needed for day-to-day functions. Some URL filtering solutions control and protect enterprises and employees from Internet threats including spyware, adware, shareware, malware, etc. The advent of new communication technologies has had tremendous impact in the growth and promotion of businesses spanning across many applications including online-banking, e-commerce, and social networking. In fact, in todays age it is almost mandatory to have an online presence.

There are a wide variety of techniques to implement such attacks, such as explicit hacking attempts, drive-by download, social engineering, phishing, watering hole, manin-the middle, SQL injections, loss/theft of devices, denial of service, distributed denial of service, and many others. Considering the variety of attacks, potentially new attack types, and the innumerable contexts in which such attacks can appear, it is hard to design robust systems to detect cyber-security breaches. The limitations of traditional security management technologies are becoming more and more serious given this exponential growth of new security threats, rapid changes of new IT technologies, and significant shortage of security professionals. Most of these attacking techniques are realized through spreading compromised URLs (or the spreading of such URLs forms a critical part of the attacking operation)

II. LITREATURE SURVEY

1. Rasula et al. have developed an algorithm for calculating trust score for each user in heterogeneous social graph for Twitter. The trust score is special a feature that can be used to detect malicious activities in Twitter with high accuracy. Their classifier attains an improved measure is 81 percent and with an accuracy of 92.6 percent. They have successfully detected malicious users. For calculating trust score they have considered only short URLs of trending topics. Based on the backward propagation, they assign trust score to tweets if trending topics present in that tweet and followed by the users. Future work deals with calculation of trust score by considering the short URLs present in the tweet.

2. Kurt Thomas et al. developed a system Monarch which is a real-time system for filtering scam, phishing, and malware URLs as they are submitted to web services. He showed that while Monarchs architecture generalizes to many web services being targeted by URL spam, accurate classification hinges on having an intimate understanding of the spam campaigns abusing a service. In particular, he showed that email spam provides little insight into the properties of Twitter spammers, while the reverse is also true. He explored the distinctions between email and Twitter spam, including the overlap of spam features, the persistence of features over time, and the abuse of generic redirectors and public web hosting.

3. Peter Likarish et al. the World Wide Web expands and more users join, it becomes an increasingly attractive means of distributing malware. Malicious javascript frequently serves as the initial infection vector for malware. He train several classifiers to detect malicious javascript and evaluate their performance. He proposed features focused on detecting obfuscation, a common technique to bypass traditional malware detectors. As the classifiers show a high detection rate and a low false alarm rate, he proposed several uses for the classifiers, including selectively suppressing potentially malicious javascript based on the classifiers recommendations, achieving a compromise between usability and security.

4. Doyen Sahoo et al. performed a survey that the malicious website, is a common and serious threat to cybersecurity. Malicious URLs host unsolicited content (spam, phishing, drive-by exploits, etc.) and lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation), and cause losses of billions of dollars every year.It is imperative to detect and act on such threats in a timely manner. Traditionally, this detection is done mostly through the usage of blacklists. However, blacklists cannot be exhaustive, and lack the ability to detect newly generated malicious URLs.

5. De Wang et al. implemented a spam detection framework to detect spam on multiple social networks. Through the experiments, he show that his framework can be applied to multiple social networks and is resilient to evolution due to the spam arms-race. In the future, he plan on testing and evaluate the framework on live feeds from social networks.

III. EXISTING SYSTEM

URL shortening is used to create shorter aliases for long URLs. We call these shortened aliases short links. Users are redirected to the original URL when they hit these short links. Short links save a lot of space when displayed, printed, messaged, or tweeted. Additionally, users are less likely to mistype shorter URLs.

Machine Learning Approaches. These approaches try to analyze the information of a URL and its corresponding websites or webpages, by extracting good feature representations of URLs, and training a prediction model on training data of both malicious and benign URLs. There are two-types of features that can be used - static features, and dynamic features. In static analysis, we perform the analysis of a webpage based on information available without executing the URL (i.e., executing JavaScript, or other code). The features extracted include lexical features from the URL string, information about the host, and sometimes even HTML and JavaScript content. Since no execution is required, these methods are safer than the Dynamic approaches. The underlying assumption is that the distribution of these features is different for malicious and benign URLs. Using this distribution information, a prediction model can be built, which can make predictions on new URLs.

The goal of machine learning for malicious URL detection is to maximize the predictive accuracy. Both of the folds above are important to achieve this goal. While the first part of feature representation is often based on domain knowledge and heuristics, the second part focuses on training the classification model via a data driven optimization approach. Illustrates a general work-flow for Malicious URL Detection using machine learning. The first key step is to convert a URL u into a feature vector x, where several types of information can be considered and different techniques can be used. Unlike learning the prediction model, this part cannot be directly computed by a mathematical function (not for most of it). Using domain knowledge and related expertise, a feature representation is constructed by crawling all relevant information about the URL. These range from lexical information (length of URL, the words used in the URL, etc.) to host-based information (WHOIS info, IP address, location, etc.).

1. **Bitly:** Best URL shortener for businesses branding and tracking links Bitly is a full service, business-grade URL shortener, although if your needs are modest, you can also use it anonymously to shorten long URLs and be on your way. But it stands out for its business offering. Part of the appeal is that Bit.ly is simple and easy to use. It has a comprehensive dashboard where you can track statistics about your links, such as click-through rates, geographic data of people visiting your links, and so forth. Tools for tracking campaigns are easy to use as well. With Bitly free limited account, you can customize your shortened URLs, track click rates, and get information about your top referrers, but only for 500 branded links and 10,000 non-branded links. It a generous free plan and could very well be adequate for some small businesses. Enterprisegrade accounts (custom pricing) allow you to make as many branded links as you want, plus see more data in reports about who clicks your links. Bitly is the best URL shortener for large businesses looking to brand and track links, and its a great choice for small businesses that want to generate short URLs and follow their stats for a modest number of campaigns.

2. **TinyURL:** Best URL shortener for quick, anonymous use Free URL shortener TinyURL has been in the game since 2002, and for good reason. Its a wonderful tool when you need to create a short link in a hurry that will never expire. TinyURL can suggest a shorter URL for you, or you can customize the result, although it will start with tinyurl.com/. TinyURL also offers a toolbar button that lets you generate a short link from the current webpage on screen. Its a little different from a typical browser plugin. On Tiny URLs main page, there are instructions to drag a link from the page into your toolbars links section. That link is actually a little script. From any web page, you can click that link and it will take you back to TinyURL where a shortened link will have already been generated for the page where you started. Although TinyURL is entirely free and anonymous to use, it doesnt contain any reports or information about your links and their popularity.

3. **Link :** Best URL shortener for small businesses Bl.ink is a full-featured URL shortener service that you use it to not only turn long URLs into short ones but also track the traffic coming from your links. Its dashboard shows trending links and general statistics, while an analytics page lets you dive into traffic by device, location, and referrers. You can also drill down into clicks by the time of day. Tags, which you can add to your shortened links, let you view your link traffic in new and custom ways. Bl.ink offers four tiers of paid plans, starting at Dollar 12/month, to give small businesses, teams, and enterprises a variety of options, based on the number of links you need to generate and track. Free account holders can generate 1,000 links and track up to 1,000 clicks per link. Free accounts can connect to one domain for making branded links.

4. **google:** The Google URL Shortener at goo.gl is a service that takes long URLs and squeezes them into fewer characters to make a link that is easier to share, tweet, or email to friends. Users can create these short links through the web interface at goo.gl, or they can programatically create them through the URL Shortener API. With the URL

Shortener API you can write applications that use simple HTTP methods to create, inspect, and manage goo.gl short links from desktop, mobile, or web. Links that users create through the URL Shortener can also open directly in your mobile applications that can handle those links.

This Tool follows the path of the URL It allows you to see the complete path a redirected URL goes through. It will show you the full redirection path of URLs, shortened links, or tiny URLs. Dataset Around 114,400 URLs were collected initially containing benign and malicious URLs in four categories:

Spam, Malware, Phishing and Defacement. Four singleclass datasets by mixing benign and malicious URLs and one multi- class datas by combining all four malicious URLs and benign URLs were generated for experiment . Benign URLs: Over 35,300 benign URLs were collected from Alexa top websites. The domains have been passed through Heritrix webcrawler to extract the URLs. Around half a million unique URLs are crawled initially and then parsed to remove duplicate and domain only URLs. Later the extracted URLs have been checked through virustotal to filter the benign URLs. – Spam URLs: Around 12,000 spam URLs were collected from publicly available web spam dataset in Phishing URLs: Around 10,000 phishing URLs were taken from OpenPhish website which is a repository of active phishing sites. – Malware URLs: More than 11,500 URLs related to malware websites were obtained from DNS-BH which is a project that maintain list of malware sites. – Defacement URLs: In], authors select 2500 URLs provided by Zone-H and extend the lists by adding URLs of pages reached by crawling the compromised sites up to the third level. After necessary filtration (e.g. URLs whose path is empty or equal to index.html, URLs whose domain is an IP address), Detecting Malicious URLs Using Lexical Analysis 475 they labelled 114,366 URLs as Defacement. However, for our experiment we randomly choose 45,457 URL.

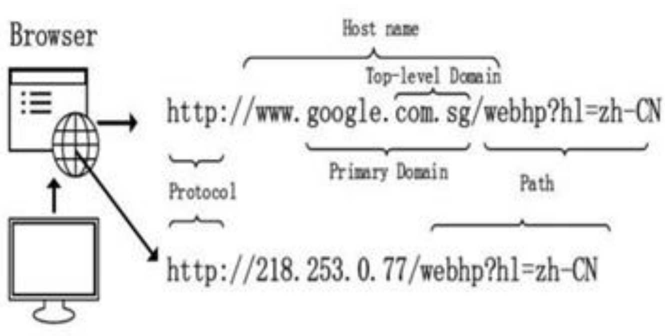


Fig. 1. Example of a URL - "Uniform Resource Locator"

PROPOSED SYSTEM ADVANTAGE

1. Identified the original url path.
2. Prevent from the malicious url
3. Protect the virus or automated malicious files

As stated earlier, the success of a machine learning model critically depends on the quality of the training data, which hinges on the quality of feature representation. Given a URL u U, where U denotes a domain of any valid URL strings, the goal of feature representation is to find a mapping : U Rd, such that (u) x where x Rd is a d-dimensional feature vector,

that can be fed into machine learning models. The process of feature representation can be further broken down. For malicious URL detection, researchers have proposed several types of features that can be used to provide useful information. We categorize these features into: Blacklist Features, URL-based Lexical Features, Host-based features, Content-based Features, and Others (Context and Popularity). All have their benefits and short comings - while some are very informative, obtaining these features can be very expensive. Similarly, different features have different preprocessing challenges and security concerns. Next, we will discuss each of these feature categories in detail.

Blacklist Features As mentioned before, a trivial technique to identify malicious URLs is to use blacklists. An existing URL as having been identified as malicious (either through extensive analysis or crowd sourcing) makes its way into the list. However, it has been noted that blacklisting, despite its simplicity and ease of implementation, suffers from nontrivial high false negatives due to the difficulty in maintaining exhaustive up-to-date lists. Consequently, instead of using blacklist presence alone as a decision maker, it can be used as a powerful feature.

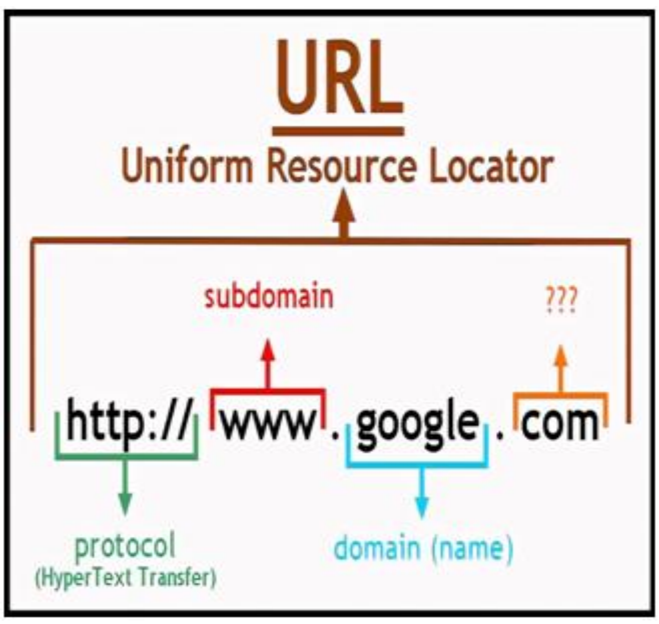
Lexical Features Lexical features are features obtained from the properties of the URL name (or the URL string). The motivation is that based on how the URL ”looks” it should be possible to identify malicious nature of a URL. For example, many obfuscation methods try to ”look” like benign URLs by mimicking their names and adding a minor variation to it. In practice, these lexical features are used in conjunction with several other features (e.g. host-based features) to improve model performance. However, using the original URL name directly is not feasible from a machine learning perspective. Instead, the URL string has to be processed to extract useful features. Next, we review the lexical features used for this task.

Host-based Features Host-based features are obtained from the hostname properties of the URL. They allow us to know the location, identity, and the management style and properties of malicious hosts. studied the impact of a few hostbased features on the maliciousness of URLs. Consequently, host- based features became an important element in detecting maliciousURLs. borrowed ideas from and proposed the usage of several host-based features including: IP Address properties, WHOIS information, Location, Domain Name Properties, and Connection Speed. The IP Address properties comprise features obtained from IP address prefix and autonomous system (AS) number.

3.1 FEATURE REPRESENTATION

Content-based Features Content-based features are those obtained upon downloading the entire webpage. As compared to URL-based features, these are ”heavy-weight”, as a lot of information needs to be extracted, and at the same time, safety concerns may arise. However, with more information available about a particular webpage, it is natural to assume that it would lead to a better prediction model. Further, if the URL- based features fail to detect a malicious URL, a more thorough analysis of the contentbased features may help in early detection of threats. The content-based features of a webpage can be drawn primarily from its HTML content, and the usage of JavaScript. categorize the content based features of a **webpage into 5 broad segments: Lexical features, HTML**

Document Level Features, JavaScript features, ActiveX Objects and feature relationships.



1. **HTML Features:** These are relatively easy to extract and preprocess. At the next level of complexity, the HTML document level features can be used. The document level features correspond to the statistical properties of the HTML document, and the usage of specific types of functionality. Propose the usage of features like: length of the document, average length of the words, word count, distinct word count, word count in a line, the number of NULL characters, usage of string concatenation, unsymmetrical HTML tags, the link to remote source of scripts, and invisible objects. Often malicious code is encrypted in the HTML, which is linked to a large word length, or heavy usage of string concatenation, and thus these features can help in detecting malicious activity.

2. **JavaScript Features:** Argue that several JavaScript functions are commonly used by hackers to encrypt malicious code, or to execute unwanted routines without the clients permission. For example extensive usage of function eval() and unescape() may indicate execution of encrypted code within the HTML. They aim to use the count of 154 native JavaScript functions as features to identify malicious URLs. [40] identify a subset (seven) of these native JavaScript functions that are often in Cross-site scripting and Web-based malware distribution. These include: escape(), eval(), link(), unescape(), exec(), and search() functions.

3. **Visual Features:** There have also been attempts made at using images of the webpages to identify the malicious nature of the URL. Most of these focus on computing visual similarity with protected pages, where the protected pages refer to genuine websites. Finding a high level of visual similarity of a suspected malicious URL could be indicative of an attempt at phishing. One of the earliest attempts at using visual features for this task was by computing the Earth Movers Distance between 2 images. Addressed the same problem and developed a system to extract visual features of webpages based on text-block features and imageblock features (using information such as block size, color, etc.).

4. Other Content-based Features: Argued that due to the powerful functionality of ActiveX objects, they can be used to create malicious DHTML pages. Thus, they tried to compute frequency for each of eight ActiveX objects. Examples include: Scripting File System Object” which can be used for file system I/O operations, Script Shell” which can execute shell scripts on the client’s computer, and Adobe Stream” which can download files from the Internet. Try to find the identity and keywords in the DOM text and evaluate the consistency between the identity observed and

the identity it is potentially trying to mimic which is found by searching. Used the directory structure of the websites to obtain insights . Other Features Recent years have seen the growth of Short URL service providers, which allow the original URL to be represented by a shorter string. This enables sharing of the URLs in on social media platforms like twitter, where the originally long URLs would not fit within the 140 character limit of a tweet. Unfortunately, this has also become a popular obfuscation technique for malicious URLs. While the Short URL service providers try their best to not generate short URLs for the malicious ones, they struggle to do an effective job. Use context information derived from the tweets where the URL was shared. Used click traffic data to classify short URLs as malicious or not. Propose forwarding based features to combat forwarding-based malicious URLs. Propose another direction of features to identify malicious URLs - they also focus on URLs shared on social media, and aim to identify the malicious nature of a URL by performing behavioral analysis of the users who shared them, and the users who clicked on them. These features are formally called ”Posting-based” features and ”Click-based” features.approach this problem with a systematic categorization of context features which include contentrelated features (lexical and statistical properties of the tweet), context of the tweet features (time, relevance, and user mentions) and social features (following, followers, location, tweets, retweets and favorite count).

Summary of Feature Representations There is a wide variety of information that can be obtained for a URL. Crawling the information and transforming the unstructured information to a machine learning compatible feature vector can be very resource intensive. While extra information can improve predictive models (subject to appropriate regularization), it is often not practical to obtain a lot of features. For example, several hostbased features may take a few seconds to be obtained, and that itself makes using them in real world setting impractical. Another example is the Kolmogorov Complexity - which requires comparing a URL to several malicious and benign URLs in a database, which is infeasible for comparing with billions of URLs. Accordingly, care must be taken while designing a Malicious URL Detection System to tradeoff the usefulness of a feature and the difficulty in retrieving it. We present a subjective evaluation of different features used in literature. Specifically, we evaluate them on the basis of Collection Difficulty, Associated Security Risks, need for an external dependency to acquire information, the associated time cost with regard to feature collection and feature preprocessing, and the dimensionality of the features obtained.

IV. CONCLUSION

The short URLs are easy to use and remember. But when short URLs redirect to destination site there some malicious actions can be perform during this redirection of short URLs. This paper explored an approach for classifying different attack types of URL automatically as benign, defacement, spam, phishing and malware through supervised learning relying on lexical features. This technique is an addon for the blacklist techniques, in which new malicious URLs cannot be identified and efficient for analyzing large number of URLs. Selected feature sets applied on supervised classification on a ground truth dataset yields a classification accuracy of 97 % with a low false positive rate. Our prediction interval filtering experiment can also be helpful to improve classifier accuracy. In addition, it can be extended to calculate risk rating of a malicious URL after parameter adjustment and learning with huge training data. Despite random forest classification accuracy is able to identify approx. 97 % of the malicious or benign URL, by using proper SD filter we could reach up to around 99 % accuracy. As future work we are planning to develop a real time tool for computing SD filter dynamically and detection of malicious URLs.

REFERENCES

[1] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. 2014. Phishing detection based associative classification data mining. Expert Systems with Applications (2014).

[2] Farhan Douksieh Abdi and Lian Wenjuan. 2017. Malicious URL Detection using Convolutional Neural Network. Journal International Journal of Computer Science, Engineering and Information Technology (2017).

[3] Sadia Afroz and Rachel Greenstadt. 2011. Phishzoo: Detecting phishing websites by looking at them. In Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on. IEEE.

[4] Yazan Alshboul, Raj Nepali, and Yong Wang. 2015. Detecting malicious short URLs on Twitter. (2015).

[5] Betul Altay, Tansel Dokeroglu, and Ahmet Cosar. 2018. Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection. Soft Computing (2018).

[6] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS.. In USENIX security symposium.

[7] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From Throw-Away Traffic to Bots:

Detecting the Rise of DGA-Based Malware.. In USENIX security symposium.

[8] A Astorino, A Chiarello, M Gaudioso, and A Piccolo. 2016. Malicious URL detection via spherical classification. Neural Computing and Applications (2016).

[9] Alejandro Correa Bahnsen, Ivan Torroledo, Luis David Camacho, and Sergio Villegas. 2018. DeepPhish: Simulating Malicious AI. In Proceedings of the Symposium on Electronic Crime Research, San Diego, CA, USA. 15–17.

[10] Ram B Basnet, Andrew H Sung, and Quingzhong Liu. 2012. Feature selection for improved phishing detection. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer.

[11] Google Safe Browsing Transparency Report (2015). www.google.com/ transparencyreport/safebrowsing/

[12] Su, K.-W., et al.: Suspicious URL filtering based on logistic regression with multiview analysis. In: 8th Asia Joint Conference on Information Security (Asia JCIS). IEEE (2013)

[13] Le, A., Markopoulou, A., Faloutsos, M.: PhishDef: URL names say it all. In: Proceedings IEEE, INFOCOM. IEEE (2011)

[14] Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)

[15] Thomas, K., et al.: Design and evaluation of a real-time URL spam filtering service. In: Proceeding of the IEEE Symposium on Security and Privacy (SP) (2011)

[16] Ma, J., et al.: Identifying suspicious URLs: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM (2009)

[17] Nunan, A.E., et al.: Automatic classification of cross-site scripting in web pages using document-based and URL-based features. In: IEEE Symposium on Computers and Communications (ISCC) (2012)

[18] Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. In: Proceedings WebApps (2011)

[19] Huang, D., Kai, X., Pei, J.: Malicious URL detection by dynamically mining patterns without pre-defined elements. World Wide Web 17(6), 1375–1394 (2014) 482 M.S.I. Mamun

a. et al.

[20] Chu, W., et al.: Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In: IEEE International Conference on Communications (ICC) (2013)

[21] Xu, L., et al.: Cross-layer detection of malicious websites. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy. ACM (2013)

[22] Garera, S., et al.: A framework for detection and measurement of phishing attacks. In: Proceedings of the ACM Workshop on Recurring Malcode (2007)

[23] Radu, Vasile: Application. In: Radu, Vasile (ed.) Stochastic Modeling of Thermal Fatigue Crack Growth. ACM, vol. 1, pp. 63–70. Springer, Heidelberg (2015)

[24] Kevin, M.D., Gupta, M.: Behind phishing: an examination of phisher modi operandi. In: Proceedings of the 1st Usenix Workshop on Large-Scale Ex-ploits and Emergent Threats (2008)

[25] Ma, J., et al.: 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. ACM (2009)

[26] Davide, C., et al.: Prophiler: a fast filter for the large-scale detection of malicious web pages. In: Proceedings of the 20th International Conference on World Wide Web. ACM (2011)

[27] Xiang, G., et al.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. (TISSEC) 14(2), 21 (2011)