A Survey on Phishing Detection Techniques

Aadil Salim Ansari [#1], Prof. Ujwala M. Patil[#2],

[1] *Department of Computer Engineering*
R.C.Patel Institute of Technology
Shirpur, Dist.Dhule, Maharashtra, India
aadil.ansari8@gmail.com
[2] *Department of Computer Engineering*
R.C.Patel Institute of Technology
Shirpur, Dist.Dhule, Maharashtra, India
patil_ujwala2003@rediffmail.com

Abstract— **The Convenience of online commerce has been willingly accepted by consumers and criminals alike. Phishing, is the act of stealing personal information via the internet for the purpose of committing financial fraud, has become a significant criminal activity on the internet. There has been good increase in identifying the threat, educating businesses and customers, and identifying countermeasures. However, there has also an increase in attack diversity and technical sophistication by the people conducting phishing fraud. Phishing has a negative impact on the economy through financial losses experienced by businesses and consumers, along with the adverse effect of decreasing consumer confidence in online commerce. Phishing scams have flourished in recent years due to favorable economic and technological conditions. The Technical resources needed to execute phishing attacks can be readily acquired through both public and private sources. Some technical resources have been streamlined and automated, allowing use by non-technical criminals. This makes phishing both economically and technically viable for a larger population of less sophisticated criminals. In this paper, different schemes to automatically detect phishing URLs by mining and extracting Meta data on URLs from various Web services are studied**

Keywords— **fraud, phishing detection, phishing URL, meta data, web services.**

I. INTRODUCTION

Phishing is a process that occurs in three steps: Planning, Attack, and Fraud. Each step is described as follows

• *Planning*

In this phase the victim is determined by the attacker to attack. Obtaining the information from the victim and how to obtain this information. Social engineering techniques are employed to gain information about the target victim. Various media, for example phone, instant messaging, clients, email, and the Internet, can be used to gain this information.

• *Attack*

Attack phase involves delivery of the phishing message and luring the victim to give up his/her credentials. Email is a popular method used to deliver the phishing message to the target.

• *Fraud*

The final step of the attacker is fraud. The attacker uses the information obtained in the attack phase to buy goods, steal money from the victims account and identity theft. This process does not stop after one attack. It is a continuing process wherein the attacker repeats the same steps with another unsuspecting victim [1]. Phishing is a social engineering crime generally defined as impersonating a trusted third party to gain access to private data. For example, an adversary might send the victim an email directing him to a fraudulent website that looks like a page belonging to a bank. The adversary can use any information the victim enters into the phishing page to drain the victim's bank account or steal the victim's identity. Phishing is a form of identity theft that occurs when a malicious Web site impersonates a legitimate one in order to acquire sensitive information such as passwords, account details, or credit card numbers. These actions include, but are not limited to, submitting personal information to the page. Though there are several anti-phishing software and techniques for detecting potential phishing attempts in emails and detecting phishing contents on websites, phishers come up with new and hybrid techniques to circumvent the available software and techniques. The Web has become a platform for supporting a wide range of criminal enterprises such as spam advertised commerce (e.g., counterfeit watches or pharmaceuticals), financial fraud (e.g., via phishing or 419-type scams) although the precise commercial motivations behind these schemes may differ, the common thread among them is the requirement that unsuspecting users visit their sites. These visits can be driven by Web search results ,email or links from other Web pages, but all require the user to take some action, such as clicking, that specifies the desired Uniform Resource Locator (URL) .

In a sense definition of phishing is closer to "web forgery," the phrase used in the Firefox user interface, than the traditional definition of phishing. This definition certainly covers the typical case of phishing pages displaying graphics relating to a financial company and requesting a viewer's login credentials. This definition also covers phishing pages which display a trusted company's logos and request that the viewer download and execute an unknown binary. Sites which claim to be able to perform actions through a third party once provided with the viewer's login credentials meet this broader

definition as well. Phishing attacks are growing rapidly by the day.

The Anti-Phishing Work Group detected a total of 27,221 unique phishing URLs in January 2013 [10]. Sophos, an anti-virus company, claims that freely downloadable do-it-yourself phishing kits exist. Consequently anyone surfing the web can now get their hands on these kits and launch their own phishing attack. These kits are supposed to contain all the graphics, web code and text required to construct bogus web sites designed to have the same look-and-feel as legitimate online banking sites. They also include spamming software which enables potential fraudsters to send out hundreds of thousands of phishing emails as bait for potential victims. These numbers and technology indicate the need for improved phishing detection and prevention and also a need for increased awareness amongst the target masses. In a typical phishing attack, scammers or "phishers" induce unsuspecting Internet users to click on a link – normally obfuscated – to their phishing websites to trick into revealing their private information, e.g., username, password, bank account, credit card number, etc. Blacklisting is the most common technique used by all major web browsers –Internet Explorer, Firefox, Chrome, Opera, etc. When a user tries to load a URL that is in the browser's blacklist, she is warned about the potential danger of visiting the webpage. Though blacklisting can be very effective in blocking the previously known phishing URL, it can miss the brand new (zero-day) phishing web pages [2]. A wide range of criminal enterprises such as spam-advertised commerce financial fraud via phishing and as a vector for propagating malware are supported by the Web. Although the precise commercial motivations behind these schemes may differ, the common thread among them is the requirement that unsuspecting users visit their sites. These visits can be driven by email, web search results or links from other web pages, but all require the user to take some action, such as clicking, that specifies the desired Uniform Resource Locator (URL) [3].

Most of the researchers argue that in order to provide a proactive protection, the machine learning classification engine, which is typically used to maintain the blacklists at the server side, must be pushed to the client browser. This would allow new URLs to be classified on-the-fly, at the time the users click on or type in the URLs [4]. Clearly, if one could inform users beforehand that a particular URL was dangerous to visit, much of this problem could be alleviated. To this end, the security community has responded by developing blacklisting services encapsulated in toolbars, appliances and search engines that provide precisely this feedback. These blacklists are in turn constructed by a range of techniques including manual reporting, honeypots, and Web crawlers combined with site analysis heuristics. Inevitably, many malicious sites are not blacklisted either because they are too new, were never evaluated, or were evaluated incorrectly (e.g., due to "cloaking"). To address this problem, some client-side systems analyze the content or behavior of a Web site as it is visited. But, in addition to run-time overhead, these approaches can expose the user to the very browser-based vulnerabilities that we seek to avoid. One of the biggest challenges of classifying URLs on-the fly, as opposed to off-line at the server side, is the latency constraint. The longer it takes to obtain the classification result of a URL, the longer a user has to wait to load that URL and the worse the user experience. Furthermore, since page loading time is a decisive factor when benchmarking web browsers, classifying URLs should not introduce high latency [5].

In reaction to increasing response from service providers and law enforcement, criminals are using increasing technical sophistication to establish more survivable infrastructures that support phishing activities. The Key building blocks for these infrastructures are the botnets that are used to send phishing emails and host phishing sites [1]. The Problem has become so severe, because of which the Internet community has put a significant amount of effort into defense mechanisms. Currently, two of the most popular services that protect the Internet users from visiting phishing sites are the Google Safe Browsing service [4] and the Microsoft Smart Screen service [6]. Both services provide client browsers with URL blacklists. The browsers, in turn, protect users from visiting the blacklisted URLs. The major problem of this protection model is that it is reactive: a phishing URL can only be included in the blacklist if it has already appeared somewhere else. Also, a recent report by the Anti-phishing Working Group (APWG) indicated more sophisticated schemes seem to have been used in phishing attacks that also exploited an increased number of brands [12]. In this paper, we have different set of heuristics that can be used in near real-time to evaluate the legitimacy of a URL. Unlike existing works in this area, the proposed heuristics are rooted in the evaluation of Meta data on URLs commonly available from search engines and other popular Web services.

The simplicity and ubiquity of the Web has fueled the revolution of electronic commerce, but has also attracted several miscreants into committing fraud by setting up fake web sites mimicking real businesses, in order to lure innocent users into revealing sensitive information such as bank account numbers, credit cards, and passwords. Such phishing attacks are extremely common today and are increasing every day. Major industry have become targets, such as financial and payment services, phishing has caused billions of dollars loss every year [5], e.g., in a spam email, or has been reported by a user. A proactive model, which can accurately identify new phishing URLs, is highly desirable to better protect the users. In this paper Section I, we precisely define "phishing" .Section II review previous work regarding anti-phishing tools and classifiers and Section III gives conclusion.

II. LITERATURE REVIEW

Basnet et al. [1] proposed a system using the heuristics and encoded each individual URL into a feature vector with 14 dimensions. They builded a classification model using Logistic Regression classifier implemented in Weka data mining framework that attempts to use these features to distinguish phishing and non-phishing URLs. Heuristics play a major role in search strategies because of exponential nature of the most problems. Heuristics help to reduce the number of alternatives from an exponential number to a polynomial number. In order to solve larger problems, domain-specific knowledge must be added to improve search efficiency. Information about the problem includes the nature of states, cost of transforming

from one state to another, and characteristics of the goals. With the exceptional growth of the Web, there is an ever escalating volume of data and information available in Web pages. There has been huge interest of researchers towards web mining. Three different research directions in the areas of web mining: web structure mining, web content mining and web usage mining. Logistic Regression (LR) [9] is a statistical model for binary classification which is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. They used Weka data mining framework for classification using Logistic regression. For some performance bottlenecks the system is not deployed in real-world phishing detection application. This system has not been used in real world phishing detection

Prakash et al. [2] proposed a system called PhishNet to identify malicious URLs using Blacklisting. It Start from known Blacklisted URLs. It grows by generating new URL variations from the original ones and then approximate matching data structure that assigns a score to each URL. Their system, PhishNet, exploits this observation using two components. In the first component, they proposed five heuristics to enumerate simple combinations of known phishing sites to discover new phishing URLs. The second component consists of an approximate matching algorithm that dissects a URL into multiple components that are matched individually against entries in the blacklist. PhishNet comprises two major components: First a URL prediction component that works in an offline fashion examines current blacklists and systematically generates new URLs by employing various heuristics (e.g., changing the top-level domains). Further, it tests whether the new URLs generated are indeed malicious with the help of DNS queries and content matching techniques in an automated fashion, thus ensuring minimal human effort. Second is an approximate URL matching component which performs an approximate match of a new URL with the existing blacklist. It uses novel data structures to perform approximate matches with an incoming URL based on regular expressions and hash maps to catch syntactic and semantic variations. They also showed that approximate matching algorithm leads to very few false positives (3%) and negatives (5%).The Problem here was it can't detect a websites which has no connection Blacklisted URL.

Le et al. [3] Proposed PhishDef, a system which implements the AROW algorithm and uses only lexical features to classify URLs. By implementing the AROW algorithm, PhishDef is able to achieve high classification accuracy even when working with noisy data, and at the same time, being lightweight in terms of both computation and memory requirement. They describe four state-of-the-art classification algorithms. These include both batch-learning (Support Vector Machine (SVM)) and online learning algorithms (Online Perceptron (OP), Confidence- Weighted (CW), and Adaptive Regularization of Weights (AROW)). A batch-based algorithm initially trains its model based on a batch of labeled data. It then uses the trained model to predict a number of new data. After some time, it re trains its model based on a new batch of labeled data. Meanwhile, an online classification algorithm continuously retrains its model upon receiving each labeled data and makes prediction of a new data using the latest updated model. Because training a model of a batch-based algorithm requires a batch of data, batch-based algorithms require significantly more memory than online algorithms. SVM constructs a hyper plane that gives the largest distance to the nearest training data points of any class. Finding this hyperplane involves solving an instance of quadratic programming. The label of a new data point is predicted by determining on which side of the hyperplane this point lies. OP suffers from a significant drawback: the update rate is fixed and does not take into account the magnitude of classification error. As a result, when making error on prediction, the model may not adapt fast enough to the change of the data, or it may make a drastic change even when the error is small. CW captures the notion of confidence in the weight of a feature. Intuitively, if the weight of a feature does not change very much over time, then one should be more confident that this weight is what it should be. AROW can be considered as a modification of CW so that the classifier is more robust in the presence of label noise. For example, if 'whitehouse.gov' is wrongly labeled as malicious (by an adversary) and fed to CW, then CW will make changes to all features that this URL has so that in the next time slot, if it sees this URL again, it will be likely to flag this URL as malicious. It can't detect a websites which has no connection Blacklisted URL.

Whittaker et al. [4] proposed a proprietary classifier to analyze millions of pages a day, examining the URL and the contents of a page to determine whether or not a page is phishing. This system can only identify a phishing page after it has been published and visible to internet users for some time. These system classifies web pages submitted by end users and collected from Gmail's spam filters. To successfully identify a wide variety of phishing pages, our system extracts and analyzes a number of features regarding these pages. These features describe the composition of the web page's URL, the hosting of the page, and the page's HTML content as collected by a crawler. A logistic regression classifier makes the final determination of whether a page is phishing on the basis of these features. The classification model used by the classifier is developed in an offline training process. The training process uses features collected by the classification system over the past three months labeled according to our published blacklist. Using our published blacklist in this fashion introduces the risk of feedback loops, where an error in our published data propagates to classification models used to generate additional published data. To minimize this risk, we also examine the relatively small number of user submitted phishing pages and reported errors manually, allowing us to break any such loops. Note that we must manually review these submissions anyway to promptly correct any user reported errors in our blacklist. However, less than one percent of the input to our system receives a manual review, leaving our automatic system to handle the bulk of the analysis. This system can only identify a phishing page after it has been published and visible to internet users for some time. Web mining can facilitate marketing patterns and tailor market to bring right products and services to right customers. It can help in making decisions in customer relationship management and also improve quality of mining.

Ma et al. [5] proposed a method to classify malicious URLs using lexical and host based properties of the URLs. Most of the features are generated by the "bag-of-words" representation of the URL, registrar name, and registrant name, binary features are also used to encode all possible ASes, prefixes and geographic locales of an IP address. The resulting URL descriptors typically have tens of thousands of binary features. The High dimensionality of these feature vectors poses certain challenges for classification. Though only a subset of the generated features may correlate with malicious Web sites, we do not know in advance which features are relevant. More generally, when there are more features than labeled examples. They used classification models i.e. Naive Bayes, Support Vector Machine (SVM) and Logistic Regression. Naive Bayes are commonly used in spam filters, this basic model assumes that for a given label, the individual features of URLs are distributed independently of the values of other features. The model parameters in the Naive Bayes classifier are estimated to maximize the joint log-likelihood of URL features and labels, as opposed to the accuracy of classification. Optimizing the latter typically leads to more accurate classifiers, notwithstanding the increased risk of overfitting. SVMs are widely regarded as state-of-the-art models for binary classification of high dimensional data. SVMs are trained to maximize the margin of correct classification, and the resulting decision boundaries are robust to slight perturbations of the feature vectors, thereby providing a hedge against overfitting. The required optimization can be formulated as an instance of quadratic programming, a problem for which many efficient solvers have been developed. They experimented with both linear and radial basis function (RBF) kernels. Problem here is that it cannot predict the status of previously unseen URLs and systems based on evaluating site content and behavior which require visiting potentially dangerous sites.

Garera et al. [6] proposed the use of 18 hand selected features to classify phishing URLs. They used their features in a logistic regression classifier that achieves a very high accuracy. They also found that it is often possible to tell whether or not a URL belongs to a phishing attack without requiring any knowledge of the corresponding page data. Identified several fine grained heuristics that can be used to distinguish between a phishing URL and a benign URL. These heuristics are used to model a logistic regression classifier. In addition to obfuscation style heuristics, which has been considered in previous work, their classifier also, incorporates several general heuristics based on Google's Index Infrastructure. They categorize features into four groups: Page Based, Domain Based, Type Based and Word Based features. Page Based include the Page Rank of a web page, its presence in the index and its quality. Page Rank is a numeric value on a scale of [0,1] that represents the relative importance of a page within a set of web pages. Phishing web pages are short lived and thus either have a very low Page Rank or their Page Rank does not exist in the Crawl Database. Three page rank features that provide discriminatory power are the Page Rank of URL, Page Rank of Host and whether the Page Rank is present in Crawl Database. Domain Based category contains only one feature, whether or not the URL's domain name can be found in the White Domain Table. Phishing URL domains are usually obfuscated (Type I, II and III) or unknown (Type IV). Word Based features of Phishing URLs are found to contain several suggestive word tokens. For example the words login and sign in are very often found in a phishing URL. The system continues to use some of the features they describe while expanding the feature set considerably. However, the original preparation of the training set involved manual labeling, which is not feasible for training sets as large as the ones.

Zhang et al. [7] proposed CANTINA, content-based approach to detecting phishing websites, based on the TFIDF information retrieval algorithm. TF-IDF is an algorithm often used in information retrieval and text mining. TF-IDF yields a weight that measures how important a word is to a document in a corpus. The Importance increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus. The Term frequency (TF) is simply the number of times a given term appears in a specific document. The Inverse document frequency (IDF) is a measure of the general importance of the term. Roughly speaking, the IDF measures how common a term is across an entire collection of documents. Thus, a term has a high TF-IDF weight by having a high term frequency in a given document (i.e. a word is common in a document) and a low document frequency in the whole collection of documents (i.e. is relatively uncommon in other documents). CANTINA focuses on developing and evaluating a new heuristic based on TF-IDF, a popular information retrieval algorithm. CANTINA not only makes use of surface level characteristics (as is done by other toolbars), but also analyzes the text-based content of a page itself. They discovered that TF-IDF yielded fairly good accuracy (correctly labeling legitimate sites as legitimate and phishing sites as phishing), but also found that it had a fair number of false positives (incorrectly labeling legitimate sites as phishing). To address this problem, developed a larger set of heuristics. Their heuristics include Age of Domain, Known Images, Suspicious URL, Suspicious Links, IP Address, Dots in URL, Forms. Age of Domain checks the age of the domain name. Many phishing sites have domains that are registered only a few days before phishing emails are sent out and used a WHOIS search to implement this heuristic. It measures the number of months from when the domain name was first registered. If the page has been registered longer than 12 months, the heuristic will return +1, deeming it as legitimate and otherwise returns -1, deeming it as phishing. Known Images checks whether a page contains inconsistent well-known logos. For example, if a page contains eBay logos but is not on an eBay domain, then this heuristic labels the site as a probable phishing page. Suspicious URL checks if a page's URL contains an "at" (@) or a dash (-) in the domain name. An @ symbol in a URL causes the string to the left to be disregarded, with the string on the right treated as the actual URL for retrieving the page. Most phishing pages contain such forms asking for personal data, otherwise the criminals risk not getting the personal information they want. He found that blacklist was competitive with other products, although their sample size was relatively small. CANTINA examines the content of a web page to determine whether it is legitimate or not. However it only can perform classification on English language.

Sr.no	Authors	Title	Year	Remark
1	Basnet et al.[1]	"Mining Web to Detect Phishing URLs".	*2012.*	They Proposed a novel approach for classifying phishing URLs or non-phishing using supervised learning across features extracted from various Web services. They used Weka data mining framework for classification using Logistic regression. This system has not been used in real world phishing detection
2.	Prakash et al. [2]	"Phishnet: Predictive blacklisting to detect phishing Attacks"	*2010*	They Proposed a system to identify malicious URLs using Blacklisting. It Start from known Blacklisted URLs. It grows by generating new URL variations from the original ones and then approximate matching data structure that assigns a score to each URL. It can't detect a websites which has no connection Blacklisted URL.
3.	Le et al.[3]	"PhishDef: URL Names Say It All"	*2010*	They Proposed PhishDef, a system which implements the AROW algorithm and uses only lexical features to classify URLs. By implementing the AROW algorithm, PhishDef is able to achieve high classification accuracy even when working with noisy data and at the same time being lightweight in terms of both computation and memory requirement. It requires overhead of querying servers.
4.	Whittaker et al.[3]	"Large-Scale Automatic Classification of Phishing Pages"	*2010*	They Proposed a proprietary classifier to analyze millions of pages a day, examining the URL and the contents of a page to determine whether or not a page is phishing. This system can only identify a phishing page after it has been published and visible to internet users for some time.
5.	Ma et al.[4]	"Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs"	*2009*	They Proposed a method to classify malicious URLs using lexical and host based properties of the URLs. Problem here is that it cannot predict the status of previously unseen URLs and systems based on evaluating site content and behavior which require visiting potentially dangerous sites.
6.	Garera et al.[5]	"A Framework for Detection and Measurement of Phishing Attacks."	*2007*	They proposed the use of 18 hand selected features to classify phishing URLs. They used their features in a logistic regression classifier that achieves a very high accuracy. They also found that it is often possible to tell whether or not a URL belongs to a phishing attack without requiring any knowledge of the corresponding page data.
7.	Zhang et al.[6]	"CANTINA: A Content Based Approach to Detecting Phishing Websites"	*2007*	They proposed CANTINA, content-based approach to detecting phishing websites, based on the TFIDF information retrieval algorithm. CANTINA examines the content of a web page to determine whether it is legitimate or not. However It only understands English language.

III. CONCLUSIONS

Malicious Web sites are a prominent and undesirable Internet scourge. To protect end users from visiting those sites, the identification of suspicious URLs is an important part of a suite of defences. However, URL classification is a challenging task because new features are introduced daily as such the distribution of features that characterize malicious URLs evolves continually. These are the different proposed approach for classifying phishing URLs or non-phishing using supervised learning across features extracted from various Web services. Though there may be some performance bottlenecks if the system is deployed in real-world phishing detection application, it can be shown that mining the Meta information on a URL across the Web can effectively detect a potentially dangerous URL and thus help Internet users from avoiding those sites.

References

[1] R. Basnet and A. Sung," Mining Web to Detect Phishing URLs", In Proc of the 2012 11th International Conference on Machine Learning and Applications - Volume 01, Washington DC,USA. pp 568-573, 2012.

[2] P. Prakash, M. Kumar, R. Kompella, and M. Gupta, "Phishnet: Predictive blacklisting to detect phishing attacks," in Proc. of 29th IEEE International Conference on Computer Communications(INFOCOM), SanDiego, USA, pp. 346–350,2010.

[3] A. Le, A. Markopoulou and M Faloutsos, "PhishDef: URL Names Say It All" In Proc. of Arxiv, Sep 2010.

[4] C. Whittaker, B. Ryner and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," In Proc. Of 17th Annual Network and Distributed System Security Symposium, California, USA, 2010.

[5] J. Ma, L. K. Saul, S. Safage and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," In Proc. of ACM SIGKDD, Paris, France. pp.1245-1253, 2009.

[6] S. Garera, N. Provos, M. Chew and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," In Proc. of 5th ACM Workshop on Recurring Malcode (WORM 07), ACM Press, New York, pp. 1-8,2007.

[7] S. Garera, N. Provos, M. Chew and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," In Proc. of 5th ACM Workshop on Recurring Malcode (WORM 07), ACM Press, New York, pp. 1-8,2007.

[8] M. Cova, C. Krueger and G. Vigna."Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code", In Proc.World Wide Conference, Releigh, North Carolina, Pp. 281-290, 2010.

[9] S. le Cessie and J. C. van Houwelingen, "Ridge Estimators in Logistic Regression," Applied Statistics, pp.191-201,1992.

[10] "Anti phishing working group." 21 Oct 2013. [Online]. Available: http://antiphishing.org.

[11] "Goole safe browsing api," 21 Oct 2013. [Online]. Available: http://code.google.com/apis/safebrowsing/.

[12]]"Phishtank," 20 Oct 2013. [Online]. Available: http://www.phishtank.com

