An Anti-Phishing Kit Scheme for Secure Web Transactions

A. A. Orunsolu[1] and A. S. Sodiya[2]

[1]*Department of Computer Science, Moshood Abiola Polytechnic, Ojere, Abeokuta South-West, Nigeria*
[2]*Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria*
orunsolu.abdul@mapoly.edu.ng, sodiyaas@funaab.edu.ng

Keywords: Backdoor, Electronic Commerce, Malicious Site, Internet, Phishing Kit.

Abstract: In this work, an anti-phishing approach was proposed against phishing pages generated by phishing kits. The architecture consists of a Sorter Module (SM) and Signature Detection Module (SDM). The SM is used to separate pages with login attributes and obfuscated scripts from other pages within the system. These sorted pages are fed into the SDM, where the signature of the suspicious page is generated. In SDM, a two-tier classifier is employed to generate phishing label based on signature analysis. Experimental results of the approach indicated a detection accuracy of 100% on specific phishing kit-generated sites and 98% on general phishing/legitimate data. To determine the detection time of the approach, latency analysis of the system was performed. The results indicated a latency 0.3s and standard deviation of 0.367s for the various operations performed by the system during detection. Thus, the approach effectively detects phishing pages by using 'fingerprints' from phishing kits.

1 INTRODUCTION

Today, there is an enormous growth of digital applications in both private and public domains. Business services and relationships are redefined as people's reliance on the internet technology continues to grow at an unprecedented rate. The proliferation of Internet has created a lot of opportunities in terms of automatic availability of services, global coverage, efficiency, reliability, and zero-delay in service delivery. Despite these noble contributions of the internet technology, the security issues of the online communication have become key concern to the stakeholders.

In the recent times, hackers have continuously managed a host of online black markets which threaten stakeholders' confidence in the usability of internet technology (Ajaya et al., 2015). This range of criminal enterprises includes spam-advertised commerce, botnet attacks, and a vector for propagating malware (Islam and Abawajy, 2013). The incidence of internet black market climaxed with the advent of phishing, in which both the service providers and online business operators have suffered consequences such as damaged reputation and huge financial losses. Also, unwary users share their own negative experiences at these malicious sites (Aparna and Muniasamy, 2015).

Despite the existence of various anti-phishing measures, the frequency of phishing incidences continues to increase (Kathryn et al., 2015). For instance, RSA's online fraud report showed estimated losses of over $5.9 billion by global organizations in nearly 450,000 attacks in 2013 (RSA, 2014). Fig 1 presents the ominous illustration of unique phishing sites detected from January to September 2015 (APWG, 2016). Generally, cyber criminals use phishing predominantly as a technique for obtaining identity related information employing both the social engineering and technical subterfuge (Han et al., 2012).

In the past, setting up a fake website with similar feel like the benign site could be achieved by copying HTML document of a website and modified them. However, the rise in the number of phishing sites may be unconnected with HTML approach as phishers now prefer exploit kits/phishing kits to the latter (RSA, 2014). These exploit kits simplify the creation of fraudulent websites by stealing the source code of legitimate web pages (Medvet et al., 2008). According to RSA online report in 2014, abundant tools and offerings have flooded the underground markets, thereby making the lives of phishers and would-be-phishers easier. While the creation of these malicious sites continues to give phishers opportunity to create fraudulent services, majority of internet users remain ignorant, unconscious, or negligent of

Orunsolu, A. and Sodiya, A.
An Anti-Phishing Kit Scheme for Secure Web Transactions.
DOI: 10.5220/0006074900150024
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 15-24
ISBN: 978-989-758-209-7

the adverse effects of phishing (Kathryn et al., 2015). Although a vast number of literatures identified the unpopular consequences of phishing toolkits, there is paucity of literature that concentrates on how to counter phishing from exploit kits perspective.

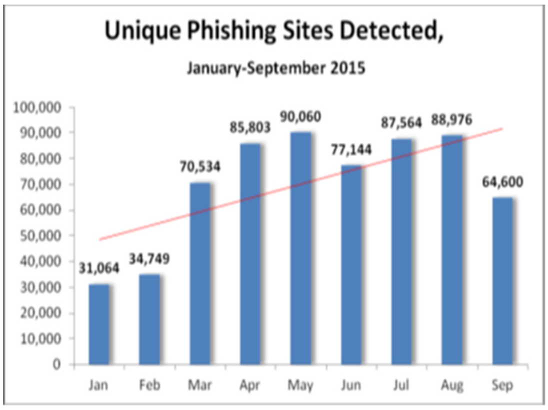

Figure 1: Phishing Volume from JAN-SEPT 2016 (APWG, 2016).

To this end, we report an anti-phishing technique based on the features of phishing kits as reported in some extant literature (McCalley et al., 2011; Cova et al., 2008). The rationale behind the design is simple. The proliferation of phishing campaign has been attributed to the availability of phishing kit to hackers who employ the tool with minimal efforts (Ajaya et al., 2015). The proposed technique creates an arm race between the hackers who employ the tool to create phishing pages and the approach which disrupts the efficacy of the tool by flagging such page as fake. In addition, the approach will reduce the degree of trust associated with phishing toolkits since there is now a defense model that targets their architecture. This will make hackers spend time checking the efficiency of kit before their deployment. Moreover, on the part of phishing kits authors, our design presents an attacking model for which they must consider defense during their design. We affirm that this is an advantage since attack is the best form of defense.

The rest of the paper is organized as follows: Section 2 presents related works, where our approach is compared with other existing anti-phishing techniques. The overall architecture and design details of our proposed methodology are discussed in Section 3. In Section 4, the implementation and the evaluation of the proposed method are presented. Conclusions and future works are presented in Section 5.

2 RELATED WORKS

Anti-phishing research has attracted a lot of interests from security experts from both academics and IT industries. Researchers have developed a plethora of countermeasures such as list-based approaches (i.e. Whitelist and Blacklist), heuristics approaches, hybrid approaches or multifaceted mechanisms (Gowtham and Krishnamurthi, 2014). Table 1 presents a summary of related works in comparison with our proposed scheme with respect to the following parameters: Kit defamation, drop email discovery, Client independent (i.e. client-side vulnerabilities) and search engine independence

A number of studies have examined the reasons that people fall for phishing attacks. For instance, Dhamija et al., 2006 identified lack of computer system knowledge, lack of knowledge of security and security indicators, visual deception and bounded attention. The authors further showed that a large number of people cannot differentiate between legitimate and phishing web sites, even when they are made aware that their ability to identify phishing attacks are being tested. In a similar vein, Sheng et al. investigated the demographic analysis of phishing susceptibility. Their works showed that women were more susceptible to phishing than men (Sheng et al., 2010).

Table 1: Comparison of related works with our approach.

Work	Kit Defamation	Drop e-mail discovery	Client independence	Search Engine independence
Aparna et al.	No	No	Yes	Yes
Shahriar et al.	No	No	No	No
Han et al	No	No	Yes	No
Olivo et al	No	No	No	Yes
Gowtham et al	No	No	Yes	No
Our work	Yes	Yes	Yes	Yes

In a more recent study, Mohammed et al., 2015. conducted user study with the use of eye tracker to obtain objective quantitative data on user judgment of phishing sites. Their results indicated that users detected 53% of phishing sites even when primed to identify them with little attention on security indicators.

A new method based on hybrid approach which depends on profiling phishing attacks has offered significant progress in the quest against phishing. Islam et al., 2013 investigated a three-tier classification approach to detect phishing emails where accuracy of detection is up to 97%. However, this technique suffered from lengthy training time and complex analysis. In a similar vein, Gowtham et al., studied the characteristics of legitimate and phishing pages by proposing heuristics, which are characteristics that are found to exist in phishing attacks. The authors extracted 15 features as heuristics for evaluating the phishiness of a webpage. Before applying heuristics to the webpages, the proposed system used two preliminary screening modules to reduce superfluous computation in the system. However, this approach suffered from client-side exploits such as Java exploit attacks and high categorization time (Gowtham and Krishnamurthi, 2014).

Cova et al., 2008 presented an interesting research on analysis of phishing kits. The authors demonstrated that there is no such thing as a free phishing kit in the underground economy. This was based on the analysis of a large number of kits. The authors found that the kit authors developed backdoors in their kits using obfuscated code. These backdoors were used to send a copy of information collected by the inexperienced kit users to third parties. In the same vein, McCalley et al., 2011 analyzed a "back-doored" phishing kits distributed by the infamous Mr-Brain hacking group. The authors showed a number of obfuscated codes used by the kit creator that allowed a third party to access the credentials of internet victims. In our work, we used the analysis of these works to present a defense framework against phishing kits.

Larcom et al., 2006 and Larson (2010) considered the instrument of Law in the race against phishing.

Larcom et al., 2010 recommended that courts should consider either large-scale damages against individual phishers or secondary liability against internet service providers (ISP) under the areas of either intellectual property (IP) or unfair competition law.

Our approach offers the following contributions to anti-phishing research:

1. We propose an interesting angle to detect phishing websites based on the analysis of phishing kits and some other generalized features of a typical phishing attacks.
2. We evaluate the proposed approach using phishing pages created by a numbers of individuals.
3. We evaluate the system for both accuracy and performance

3 THE PROPOSED SYSTEM

The research reported in this paper is focused on phishing detection using the analysis of phishing kits. Phishing kits are usually distributed as .zip, .gz, .tar, or .rar archives that contain two types of files. The first file displays a copy of the targeted web site and the second file contains the scripts used to save the phished information and send it to phishers. Phishing kits are distributed on websites with instructions for the phishers to insert their email address in order to receive the stolen information. One of the distinguishing features of kit-weaved sites is drop email address usually passed to the PHP ”action” file. In certain cases, the drop email address is obfuscated into a hexadecimal form (Cova et al., 2008; McCalley et al., 2011). In general, majority of phishing kits contains:

a. Resources to replicate target site using HTML pages, JavaScript and CSS files, images and other media files [3,11]
b. Automated email address field to obtain and validate victim's information e.g. PHP getmxrr ( ) function [3].
c. Obfuscation techniques for hiding drops and backdoor within the phishing kit

From the foregoing, we present the overall system design of the anti-phishing kit architecture. In order to design an effective architecture, a two-stage anti-phishing service is created into a single workflow. These services consist of: (i) a Sorter Module (SM) and (ii) a Signature Detection Module (SDM). The SM retrieves the suspicious loading sites and checks for presence of login form. This is because; the primary motivation of phishing attacks is to fraudulently obtained users' credentials using spoofed sites. The SM performs the second check for obfuscation detection. Obfuscation is a technique use to hide attacks from static detection by causing the appearance of malicious string to change. In this way, the code evades detection tools. If no login form or obfuscated code is observed in the source code by

SM, the loading page is allowed to continue its session. However, if these traits are observed on the site, the SM transferred the page to the SDM. In SDM, the suspected page is processed further. Here, the features of the pages are extracted and their signatures are obtained. In this work, 18 heuristic features are considered. From the viewpoint of phishing kits, heuristics such as drops, hexadecimal scripts etc. are referred to as Third party heuristics. To improve the performance of signatures on general phishing datasets, the URL characteristics and keyword identity. Then, all these signatures are used by hybrid classifier to generate appropriate label for the loading sites. The hybrid classifier is optimized using machine learning algorithms consisting of Naïve Bayes and Support Vector Machine.

4 COMPONENTS OF ANTI-PHISHING KIT SYSTEM

The proposed Anti-Phishing kit (APK) methodology consists of the Sorter Module (SM) and Signature Detection Module (SDM) as depicted in Figure 2. These two components are used to analyze each webpage visited and determine the status of the page with phishing kit. To detect kit generated sites, the APK loads the webpage, w, a user wants to visit, and scan through the source file of w for various inherent phishing kit signatures and the other relevant anti-phishing signatures. We hope you find the information in this template useful in the preparation of your submission.

4.1 The Sorter Module

For a given webpage *w*, the SM component of APK first identifies the presence of login fields. This is to reduce superfluous computation on webpages without login fields as most phishing websites are meant to steal user credentials. Generally, the presence of login form on a page is usually characterized by presence of Form tags, Input fields, and login keywords (e.g. password, PIN, ID, username, social security number, account number etc.). The Input fields usually hold user input and login attributes which distinguish a login form from other types of forms.

The SM employs Latent Dirichlet Allocation (LDA) to manage the three login form properties. This is due to the sensitivity of LDA to changes in keywords usage which make it good for handling synonyms. For instance, a phisher may replace the word ‘password’ with ‘secret word or phrase’ to circumvent login detection process. LDA posits that one way of sorting the content of w is to look at the set of words it uses. Because words carry very strong semantic information, webpages that contain similar content i.e. set of login words, will most likely use a similar set of words. As such, mining an entire corpus of webpage can expose sets of words that frequently co-occur within webpages. These sets of word may be interpreted as topics. The modeling process of LDA can be described as finding a mixture of topics z for each website w i.e. $P(z|w)$, with each topic described by terms t following another probability distribution, i.e., $P(t|z)$.

This formalization is given in equation 1

$$\mathrm{P}(\mathrm{t_i}|\mathrm{w}) = \sum_{\mathrm{j}=1}^{\mathrm{Z}} \mathrm{P}(\mathrm{t_i}|\mathrm{z_i} = \mathrm{j})\, \mathrm{P}((\mathrm{z_i} = \mathrm{j}|\mathrm{w}) \qquad (1)$$

Where $P(t_i|w)$ is the probability of the i*th* term for a given w and z_i is the latent topic. $P(t_i|z_i = j)$ is the probability of t_i within topic j. $P(z_i = j|w)$ is the probability of picking a term from topic j in the webpage. The number of latent topics, in the case of login attributes, has to be defined in advance. In this way, LDA estimates the topic-term distribution $P(t|z)$ and the webpage-topic distribution $P(z|w)$ from an unlabeled corpus of documents using Dirichlet priors for the distributions and a fixed number of topics (Ralf et al., 2009).

The functionality of SM extends to detecting presence of obfuscated code in the loading page. The work of Cova et al and McCalley showed that most phishing kits employed obfuscation to hide their malicious activities. In the light of this, we adopt the work of Xu et al., (2013) to perform the detection of obfuscated by SM subsystem due to its lightweight attribute.

4.2 The Signature Detection Module

To determine whether a sorted page is toolkit generated or not, the Signature Detection Module is invoked. The main heuristics used by SDM in classifying a website are the signatures extracted from the analysis of phishing kit code (called Third party heuristics), URL characteristics and keyword identity. Third party heuristics are extracted features known of a typical phishing kit. For instance, drops and hexadecimal code are used to hide detection of kits features (Figure 3). These scripts are inserted by phishers to obtain undue advantage from the users of the kit. In addition, these features retain some information about the kit used. The following section provides discussion on these heuristics.

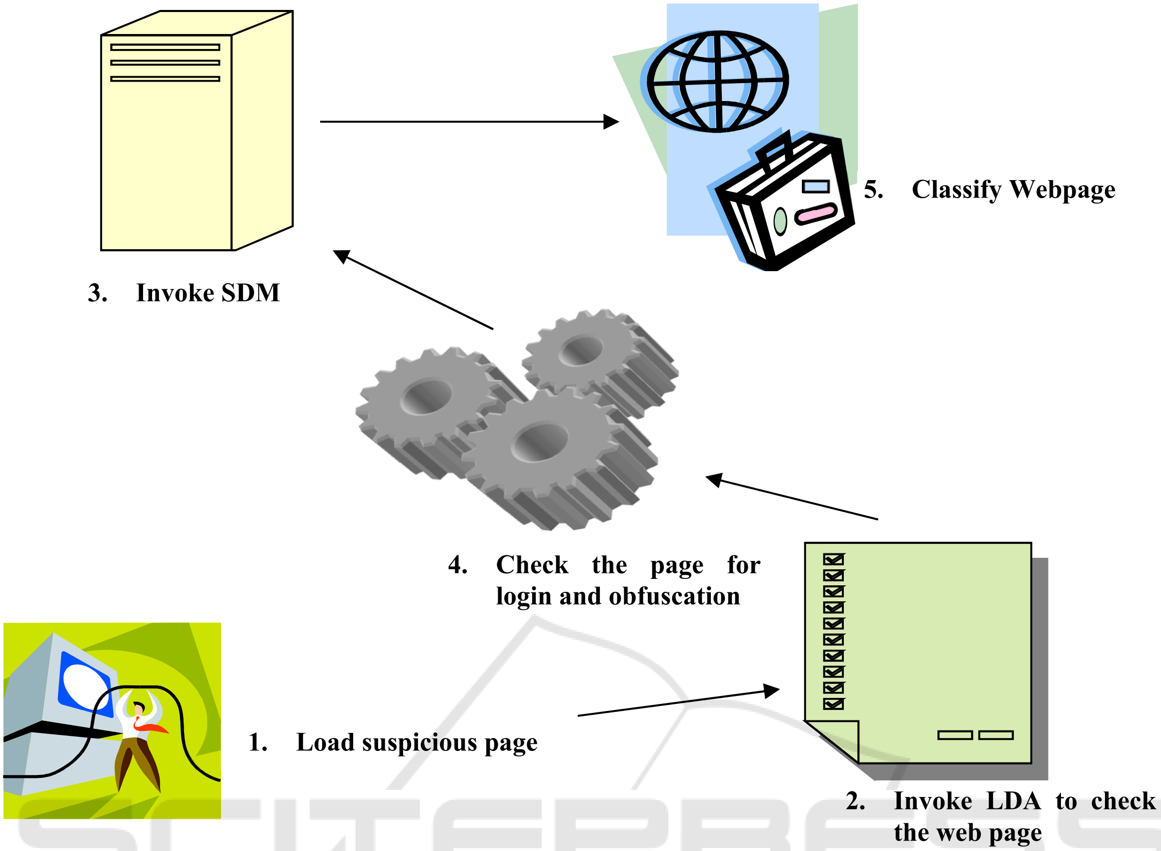
5. Classify Webpage
3. Invoke SDM
4. Check the page for login and obfuscation
1. Load suspicious page
2. Invoke LDA to check the web page

Figure 2: Anti-Phishing Kit defense architecture.

1. **Drop Attributes/Bad Forms:** Phishing kits creators usually include backdoors in their scripts. These backdoors consists of hidden drops, which covertly send victims' information to addresses different from that of kit users. Therefore, the presence of hidden drops is an indication that a page is kit-related.
2. **Hexadecimal Character Codes/Address Obfuscation:** One distinguishing feature of phishing kits is to obfuscate email addresses within a message or website. Hexadecimal codes are used to obfuscate the planted backdoors in the kits (Cova et al., 2008). If a website contains obfuscate address which is recover with instrumentation or set of command, then the page is phishing kit-related.
3. **Name of Toolkit:** Some toolkits usually add the name of the tool used to fake a site in the source code of the website as comment e.g. <!-- created with -->. The name can be found at the top of the HTML document or the copyright section at the footer of the webpage as comment which is not visible to users. Therefore, if a page contain a name not related to its content, then it is kit-based.
4. **Blank Redirection Page:** Some phishing toolkits create a blank page as the first page which will redirects to the fake page. The blank page contains some information belonging to the kit used in the source code of the blank page. If a site contains a black redirection page with information unrelated to its content, then it is kit-based.
5. **URL of Toolkit:** Apart from the toolkit name that can be found in faked sites, the URL of the toolkit used can also be found. This is to link new users to the toolkit in order for them to download it. The URL is usually commented out so that it will not be visible to web users. If a page contains out-of-comment URL not related to the page, then it is kit-based.

Keyword extraction heuristics are particularly important in identifying target organization of a phishing kit. The keyword identity set of a webpage is usually extracted by considering Document Object Model tree. The keyword extraction in SDM adopts the Term Frequency-Inverse Document Frequency (TF-IDF). The TF-IDF is a numerical statistic which reflects how important a feature is to a document in a corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a feature appears in the document, but is offset by the frequency of the feature in the corpus.

It is interesting to note that the base of the log function does not matter and constitutes a constant multiplicative factor toward the overall result. Thus, a term *t* has a high TF-IDF weight by having a high term frequency in a given document *D* (i.e. a feature is common in a document) and a low document frequency in the whole collection of documents (i.e. is relatively uncommon in other documents).

The Keyword extraction heuristics used in our proposed system are discussed as follows:

6. **Domain Name Credibility:** The domain name credibility feature determines the genuineness of the target organization by phishing kit creators using Google's PageRank system. If kit contains file for one or more target organization, then the rank of the hosting domain is compared with a threshold value (usually 5 on 0 to 10 scale) indicating the legitimacy of the site.
7. **Domain Name Identity:** Most of the website domain names have relationship to their contents. The keywords in this domain name are usually part of the base domain URL. If the keyword identity set of a page is not related to its contents, then it is phishing. Otherwise, it is legitimate and non-kit related.
8. **Out-of-Position Brand Name:** Legitimate sites often put their brand name into their domain name. On the other hand, phishing sites are always hosted on compromised or newly registered domains. If the domain keywords are not related to its brand, then the page is suspicious.
9. **Age of Domain:** This feature checks the age of the domain name. Many phishing pages claim the identity of known brand which has relatively long history. If the age of domain does not correspond to the WHOIS lookups, then it is likely to be deceptive.

The URL identity of a webpage is determined by analyzing the patterns from its hyperlinks structure. In a legitimate website most of the links points to its own domain or associated domain, but in phishing sites (including the kit-based phish sites) most of the links point to foreign domain to imitate the behavior of a legitimate page. For URL identity extraction, the SDM consider the "href" and "src' attributes of the anchor links, particularly <a>, <area>, <link>, <img>, and <script> tags from the DOM tree of a webpage. For each anchor, the SDM extracts the base domain portion from the URL, and then calculate the number of times each base domain appears. The base domain that has the highest frequency will be the URL identity of the webpage. This step is necessary in determining the behavior of the URL embedded in a suspicious webpage. In the end, the following features are considered from URL identity to generalize the detection accuracy of the proposed system:

```
$hostname = gethostbyaddr($ip);
$message = "Chase Bank Spam ReZulT\n"; ...
$message .= "User ID : $user\n";
 $messege .= "hostip" $message .= "Full Name :
     $fullname\n"; ...

 $message .= "City : $city\n";
 $messege .= "port";
 $message .= "State : $state\n"; ...
 $message .= "Mother Maiden Name : $mmn\n";
 $messege .= "@"; ...
 mail($to,$subject,$message,$headers);
      mail($messege,$subject,$message,$headers);
```

Figure 3: Sample of Drop Email Code (Cova et al., 2008).

10. **URL of Original Site:** Most phish sites usually put the URL of the original site faked as comment at the top of the html page. This is to show where the website was copied from. If such feature exists on a page, then it is phishing and possibly kit-based.
11. **Presence of User-info in the Domain Name:** In this feature, the presence of @ or dash (-) is checked for within the URL. If such feature is found, then the page is a phish site.
12. **IP Address Behavior (Either Irreversible or Reversible):** In this feature, the system checks whether the URL address of a website is a permanent IP address which does not have DNS entries. In most phishing site, the practice is usually an IP address-based URL because of its low cost. Therefore, if such feature exists, then the page is a phish site.
13. **Number of Dots in the URL:** This feature counts the number of dots in the URL as most phishing pages tend to use more dots in their URLs. If this feature exist on a page, then it is a phish site.
14. **Domain Name in the Path of the URL:** This feature checks for the presence of dot separated domain or host name in the path part of the URL. If this feature exists in a page, then it is a phish site.
15. **Presence of Foreign Anchors:** This feature examines of foreign anchors in a webpage. If a

page contains too many foreign anchors then it is likely to be deceptive.

16. **Cookie Domain:** This feature checks the transmission of text data by a web server to a web client. This text data is called HTTP cookies which are used for maintaining information about client users. If a website has a domain cookie which is in a foreign domain, then it may be deceptive as most legitimate websites have their own domain cookies or no cookies.
17. **Port Number Behavior:** This feature compares the port number part of a domain name with the stated protocol part of a URL. If the protocol does not match the port number, then the page is a phish site.
18. **SSL Protected:** Secure login pages of benign sites often have an SSL certificate while most phishing sites do not. This feature examines the certificate of a webpage and whether is it issued by trusted certificate authority. If the page's claimed identity does not appear in the attached certificate, then the page is likely to be phishing and kit-related.

The algorithm in Figure 4 presents the main structure of the proposed system.

```
Input: Web page W, Anti-phishing signatures S
Output: Phishiness level of W
   Begin
1. Load and parse page W
2. Generate the DOM from HTML of W
3. Check if W ∋ FORM input
4. Check if W ∋ Obfuscated JavaScript code
5. If (3) .OR. (4) ∈ W, invoke SM for preprocessing
else Exit // .OR. is a disjoint relation
6. Else if (5) is present, extract signature of W ∋
preprocessed DOM (W)
7. Extract URL heuristics, Keyword extraction
heuristics and Third-party heuristics from W
8. Send the signature (W) to NB-SVM classifier
trained with dataset S
9. Display the status of W
10. End if
    End
```

Figure 4: Anti-Phishing Kit Defense Algorithm.

To detect the status of the analyzed webpage, the SDM uses a hybrid classifier consisting of Naïve Bayes and Support Vector Machine based on the extracted features. The SDM uses Naïve Bayes (NB) as vectorizer and Support Vector Machine as classifier. The main problem associated with using SVM as classifier is the effort needed to transform text data into numerical data which is sometimes termed as "vectorization". It is natural to use the NB as the vectorizer for classifier based on the vector space model, such as SVM, which typically requires preprocessing to vectorize the raw text documents into numerical values. In this way, NB is used as a pre-processor for selected features in the front end of the SVM to vectorize corpus before the actual training and classification are carried out. The main procedure of the proposed hybrid classifier is described in Figure 5.

```
Algorithm: A hybrid classifier algorithm – create an
ensemble of classifiers using NB and SVM.
Input: 18-dimension feature vector space, Training
data of labeled examples S consisting of the signature
Output: Label (1: (Phished) PT related; 0: (Benign)
non-PT related)
Procedure:
1. For all signature i extracted from W
2. Compute the conditional probability of signatures
   of analyzed phishing kit given the signature of W
3. Construct the probability as input into SVM
4. Find the optimal hyper plane for signature (W)
   and signature S
5. Classify W
```

Figure 5: Two-Tier Classifier of SDM.

5 IMPLEMENTATION AND EVALUATION

Our Anti-Phishing defense system is implemented in Microsoft's Visio Studio environment on a machine with Windows 7 OS. The machine runs on an Intel Core i5 processor with 4 GB RAM and 450 GB Hard drive. We trained the algorithm using a set of web pages consisting of toolkit-generated pages and genuine pages. A preliminary test showed that this implementation can accurately detect the absence of login form on most tested sites. It loads the webpage a user wants to visit, and scans through the source file using the attributes of its keyword extraction, URL behavior and Third-party heuristics.

In order to evaluate the effectiveness of this system, we recruited the service of ethical hackers consisting of 100 students of a Computer Security class and 4 external research collaborators to create phishing pages using toolkits of their choice. In addition, phishing and legitimate data were obtained from openly available research database sources such as PhishTank, Millersmiles, Alexa ranking and

Cybercrime Archive to evaluate the performance of the system on general phishing data corpus.

We conducted three experiments to evaluate the performance of the proposed system. The first experiment and the second experiment were used to evaluate the accuracy of the proposed approach in terms of True Positive, False Positive, False Negative and True Negative. The True Positive means the actual data and predicted categories are true. The True Negative means actual and predicted categories are negative. The False Positives means the predicted should have been negative instead classified as positive. The False Negatives means predicted should have positive instead classified as negative. Accuracy is a measure of how accurate the learned system makes prediction on unseen test instances.

In the first experiment, 258 kits–generated sites created by ethical hackers were subjected to the prediction of APK. Exactly 208 of these cloned pages were generated by students of Computer Security class during a Security Lab assignment in March 2015. The remaining 50 were created by ethical hackers collaborated into this study in December 2014. Table 2 presents the number of sites created by each phishing toolkit used by the students during the cloning process. These kits perfectly faked the original sites with similar look and feel that can deceive even an experienced web user (Fig. 6). One of the phishing toolkits that are used by the ethical hackers is the HTTRACK which is an easy to use offline browser utility. This toolkit enables a phisher to download a webpage from the internet to a local directory and thereafter build recursively all directories (e.g. html files, images, link structure and other files) from the server to the phisher's computer. It is important to state that these two groups of collaborators (i.e. the computer security students and ethical hackers) did not have the knowledge of our defense system.

Table 2: Distribution of kit-created pages.

Name of Kit	Number of Webpages
Cyotek	20
A1	38
Fresh Web	30
HTTRACK	80
Webclone Maker	20
Web2Disk	20
Total	**208**

In general, these toolkits always obfuscate the links of the login-page. These toolkit-generated websites were build offline and were later put online. Then, the APK was used to download the source code of these toolkit-generated sites and checked for phishing signatures. In the end, the approach correctly labeled the 258 websites as phishing. Figure 7 presents the graphical illustration of the first experiment. APK was able to detect all the 258 websites due to the common weakness of redirection to the real web page these phishing toolkits mimicked.

Figure 6: PT-Generated First Merchants Bank Home Page.

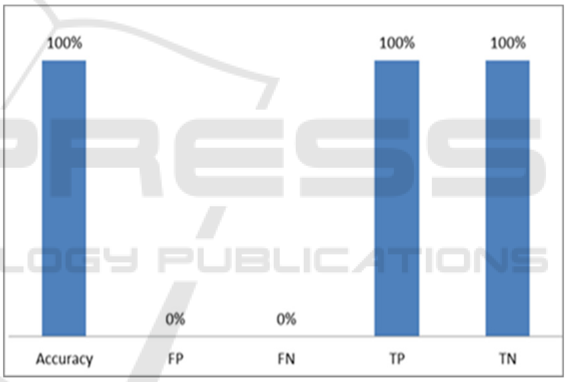

Figure 7: Experimental Results on PT-Generated Sites.

In the second experiment, the performance of the system was tested on general phishing dataset corpus. A total number of 200 phishing pages and legitimate pages were compiled over the period of 4 months from September 2014 to December 2014. Specifically, our data consists of 100 phishing pages from PhishTank (2014), Millersmiles and CyberCrime Archive (http:// cybercrime-tracker.net). On the other hand, the 100 legitimate pages were obtained from Alexa Ranking Top List which contains well-known websites with high ranking (http://www. alexa.com). These legitimate sites are popular sites which are usually target by phishers because of their large numbers of subscribers. This is easy as all the registered users of a website may not all be security conscious and sometimes, changes to operational issues of such websites may not get to all these users on time. Figure 8 showed the experimental results of the system accuracy using confusion matrix for the 200 dataset.

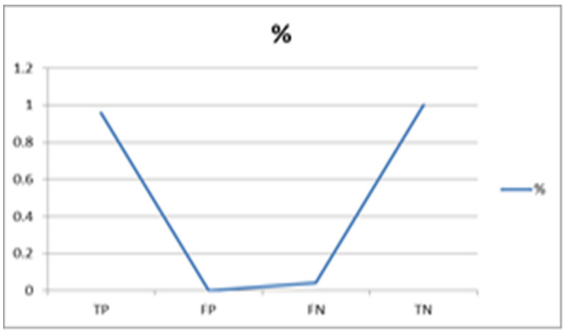

Figure 8: Experimental Results on General Data Corpus.

From the results, the system accuracy is 98% with low false negatives of 0.04%. All the correctly detected phishing sites exhibited features captured in the approach while the incorrectly labeled sites were developed with advanced features not available in phishing toolkit technology. A closer examinations of these undetected sites revealed that phishers built those sites from the scratch to escape possible detection by anti-phishing techniques.

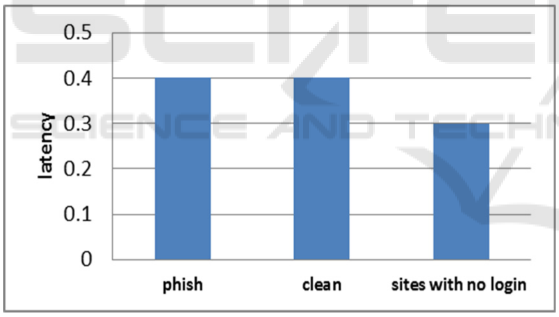

Figure 9: Average Latency Analysis of APK.

In the third experiment, the runtime analysis of the proposed system was evaluated to determine the usability issue of delay that users may experience during the detection process of APK. Using standardized timing procedure; we run experiments for different testing sites. In the process, we calculated the time between the initiation of a transaction (e.g. loading of suspicious page) to the time when the system completes the detection process. The average latency for the various operations in APK is presented in Figure 9. In our experimental analysis, we found that it took 0.3s for APK to download and check the login status of a website. The signature extraction and detection takes 0.4s and the system used 0.4s to check clean sites. The standard deviation of the system is 0.367s.

6 CONCLUSIONS AND FUTURE WORK

In this paper, the concept of phishing detection based on the analysis of phishing kit was presented and discussed. This is achieved through the Sorter Module and Signature Detection Module components of the APK architecture. The Sorter Module detects the presence of login fields and obfuscated code on a suspicious page to prevent superfluous computation in the Signature Detection Module. This is necessary because most phishing pages are set up to have access, and subsequently, steal users' data. The sorted pages with login fields and obfuscated code are sent to Signature Detection Module. In SDM, signatures are extracted from the sorted pages and its features are subsequently generated. A hybrid classifier is used to correctly label the extracted signatures. The work is implemented and evaluated using dataset from standard dataset from openly available research data sources such as PhishTank. Three experiments were conducted during the evaluation process. The first experiment, in which the performance metrics were evaluated, indicated that the accuracy of the proposed system is 100% with no false positives for specifically kit-generated sites. Whereas in the second experiment, which determines the accuracy of APK on general phishing dataset corpus from openly available data sources indicated 98% accuracy with low false positives. In the third experiment, the associated latency with the proposed system was evaluated. The evaluation results indicated a very low latency with insignificant bandwidth overhead. Future works we will determine the accuracy of the system on large datasets from specifically phishing toolkit-generated sites and open phishing dataset corpus with adequate consideration for evasion technique such as randomization in toolkits. We hope to devise a method to have access to online black market to obtain data for phishing toolkit-generated sites from professional phishers to see if the submissions of ethical hackers have any resemblance with their antics.

REFERENCES

Ajaya, N., R Luthfor, M., Nitesh, S. & Leane, H., 2015. A Multi-Modal Neuro-Physiological Study of Phishing Detection and Malware Warning. *Proceedings of CCS.*

Aparna, S. & Muniasamy, K., 2015. Phish Indicator: An Indication for Phishing Sites. *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems.*

APWG, 2016. *APWG Security Report,* s.l.: s.n.

Cova, M., Kruegel, C. & Vigna, G., 2008. There is No Free Phish: An Analysis of "Free" and Live Phishing Kits. s.l., *USENIX Workshop on Offensive Technologies*.

Dhamija, R., Tygar, J. & Hearst, M., 2006. Anti-phishing based on automated individual white-list. USA, *ACM Workshop on Digital Identity Management*.

Gowtham, R. & Krishnamurthi, I., 2014. A Comprehensive and efficacious architecture for detecting phishing pages. *Computers and Security*.

Han, W., Cao, Y., Bertino, E. & Yong, J., 2012. Using automated individual white-list to protect web digital identities. *Expert Systems with Applications*.

Islam, R. & Abawajy, J., 2013. Multi-tier phishing detection and filtering approach. *Journal of Network and Computer Applications*.

Kathryn, P. et al., 2015. The design of phishing studies: Challenges for researchers. *Journal of Computers and Security*.

Larcom, G. & Elbirt, A., 2006. Gone phishing. *IEEE Technology and Society*.

Larson, J., 2010. Enforcing Intellectual property rights to deter phishing. *Intellectual Property and Technology* .

McCalley, H., Wardman, B. & Warner, G., 2011. Analyis of Backdoored Phishing Kits. *IFIP Open Digital Library*.

Medvet, E., Kirda, E. & Kruegel, C., 2008. Visual-Similarity based phishing detection. Turkey, *Proceedings of 4th conference on Security and Privacy in Communication Networks*.

Mohammed, A., Furkan, A. & Sonia, C., 2015. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human Computer Studies,* pp. 70-82.

Ralf, K., Peter, F. & Wolfgang, N., 2009. Latent Dirichlet Allocation for Tag Recommendation. s.l., *ACM RecSys*.

RSA, 2014. Anti-Fraud Command Center, s.l.: *RSA monthly online fraud repor*t.

Sheng, S. et al., 2010. Who falls for phish? A demographic analysis of phishing susceptibility and effectivenessfor interventions. USA, *Conference on Human factors in Computing Systems*.

Xu, W., Zhang, F. & Zhu, S., 2013. *JStill:Most Static Detection of Obfuscated Malicious Javascript Code.* s.l., CODASPY.