Hindawi
Security and Communication Networks
Volume 2018, Article ID 7251750, 11 pages
https://doi.org/10.1155/2018/7251750

Research Article

Leverage Website Favicon to Detect Phishing Websites

Kang Leng Chiew, Jeffrey Soon-Fatt Choo, San Nah Sze, and Kelvin S. C. Yong

Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia

Correspondence should be addressed to Kang Leng Chiew; klchiew@unimas.my

Received 23 October 2017; Revised 10 January 2018; Accepted 30 January 2018; Published 6 March 2018

Academic Editor: Luca Caviglione

Phishing attack is a cybercrime that can lead to severe financial losses for Internet users and entrepreneurs. Typically, phishers are fond of using fuzzy techniques during the creation of a website. They confuse the victim by imitating the appearance and content of a legitimate website. In addition, many websites are vulnerable to phishing attacks, including financial institutions, social networks, e-commerce, and airline websites. This paper is an extension of our previous work that leverages the favicon with Google image search to reveal the identity of a website. Our identity retrieval technique involves an effective mathematical model that can be used to assist in retrieving the right identity from the many entries of the search results. In this paper, we introduced an enhanced version of the favicon-based phishing attack detection with the introduction of the Domain Name Amplification feature and incorporation of addition features. Additional features are very useful when the website being examined does not have a favicon. We have collected a total of 5,000 phishing websites from PhishTank and 5,000 legitimate websites from Alexa to verify the effectiveness of the proposed method. From the experimental results, we achieved a 96.93% true positive rate with only a 4.13% false positive rate.

1. Introduction

Phishing attacks can be defined as an act of deceiving victims via e-mail or a website to gain their trust to disclose their personal and financial information. With the advancement of information technology, many business agencies (e.g., banks, tourism, hotels, and airlines) can incorporate e-commerce, electronic payments, and social networking technologies into their businesses to increase sales. But this creates opportunities for phishers to gain illegal profits by disguising a wide range of services offered by financial institutions, social networking, and e-commerce websites. The Antiphishing Working Group (APWG) reported a total of 128,378 unique phishing websites detected in the second quarter of 2014 phishing activity trends report [1]. The report showed evidence that phishing activities are on the rise, which revealed that the existing antiphishing solutions were unable to resist phishing attacks efficiently.

The most common way to create a phishing website is through content replication of popular websites such as PayPal, eBay, Facebook, and Twitter. Phishing websites can be produced quickly and require little effort. This is because the phisher can simply clone the website with some modifications in the input tag to collect personal information. Furthermore, this process can be shortened by using a phishing kit [2] available on the black market. Inadvertently, advances in information technology also help phishers to develop high-profile phishing techniques to avoid phishing detectors. Figure 1 shows an example of a phishing website masquerading as PayPal. There are two flaws identified in the address bar (as shown by the red line box in Figure 1):

(i) The domain name is completely different from the genuine PayPal website.

(ii) It obfuscates the URL with HTTPS as part of the URL.

Although there are many solutions proposed to detect phishing websites, these solutions have some shortcomings. First, existing textual-based antiphishing solutions depend on the textual content of a webpage to classify the legitimacy of a website. Therefore, these solutions are incompetent to classify image-based phishing websites. A phisher can replace the textual contents with images to evade phishing detectors. Second, some phishers create phishing websites that are visually similar (e.g., webpage layout) to the legitimate website to phish potential victims. They preserve iconic images

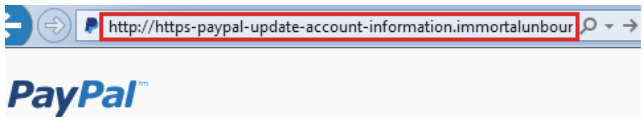

Figure 1: Example of a phishing website.

from legitimate websites to convince victims that the current webpage is benign. As a result, this type of phishing website may be incorrectly classified as legitimate by image-based antiphishing solutions that are based on similarity measurement. Third, most of the existing antiphishing solutions are unable to reveal the identity of targeted legitimate websites. Instead, they only notify the matching attributes of phishing. This can become a serious threat to Internet users if existing antiphishing solutions cannot identify the identity of new phishing website.

Our proposed method is called Phishdentity. It is driven by Chiew et al. [3], which is to find the identity of a website using the Google image search engine. In this work, we propose using the website favicon. The favicon is chosen because it represents the brand of the website. In addition, the favicon will not be affected by dynamic content (e.g., advertisement) displayed on the webpages. To determine the identity, we use the Google image search engine to return information about the favicon. The Google image search engine is chosen for this work because it allows the image (e.g., the favicon) to be used as a search query to find information. It also has the highest number of legitimate websites indexed [4]. Furthermore, using the Google image search engine can eliminate the need to maintain a database that may affect the effectiveness to detect a phishing website.

This paper is an extension of our previous paper work [5]. The previous work relies only on the usage of the favicon to identify the identity of a website. From the identity of the website, the legitimacy of the website-in-query can then be verified. However, if the website does not have a favicon, such approach will fail. In this paper, we proposed an enhanced version of the previous work with the introduction of Domain Name Amplification that further improves the accuracy of the detection and incorporation of five additional features. In particular, the additional features will be used to improve the performance in classifying websites without the presence of favicon. With the introduction of these new features, we present a more complete solution for the detection of the wide variation of phishing websites and not only the websites with the presence of a favicon as in the previous work. Furthermore, we have added a series of detailed experiments to analyze the proposed method.

The contributions of this paper are fourfold. First, we exploit the favicon extracted from the website and utilize the Google search by image engine to discover potential phishing attempts. Unlike current antiphishing solutions, our proposed method eliminates the need to perform intensive analysis on either text-based or image-based content. This has improved the detection speed. Second, the usage of mathematical equations has enabled the retrieval of the true identity from many entries of the search results. Third, we have proposed additional features to overcome the missing favicon issue. Fourth, our proposed method does not require a database of images or any presaved information from legitimate websites. Thus, it reduces the risk of having high false detection due to an outdated database.

The paper is structured as follows. The next section discusses related work. Section 3 presents the details of our approach. Section 4 describes the experiments conducted to verify the effectiveness of our solution. Finally, Section 5 concludes the paper.

2. Related Work

Many organizations, whether for-profit or nonprofit organizations, have joined forces to fight against phishing attacks. They collect and study the characteristics of phishing websites through different channels (e.g., PhishTank [6] and APWG) in order to develop effective solutions that can prevent Internet users from visiting phishing websites. In addition, these organizations also use delivery methods to educate the public about phishing websites. The delivery method involves an approach that uses posters, educational programs, campaigns, games, and so forth to convey information about phishing attacks. However, these efforts are not effective since phishing incidents are still increasing, as reported in [1, 7–9]. Therefore, it is necessary to understand the pros and cons of existing antiphishing solutions in order to develop a new solution that can overcome their limitations.

A list-based approach is a type of antiphishing solution that assesses the legitimacy of a website based on a presaved list. Normally, the list will be used as an extension of the web browser. These list-based approaches can achieve very high speeds in classification because they only compare the URL with the list. This list can be divided into whitelist and blacklist. The whitelist is a list of legitimate websites that are trusted by Internet users. Internet users can add or update the legitimate websites in the list. It is very unlikely for a whitelist to contain a phishing URL. However, Internet users can inadvertently whitelist a phishing website if they cannot distinguish between legitimate websites and phishing websites. Conversely, a blacklist contains a list of malicious websites. It is maintained and updated by the provider (e.g., Google developers). The blacklist can be very effective against phishing websites if the phishing URLs are already in the list. But the effectiveness is very much dependent on the update provided by the developer [10]. In addition, the study by Sheng et al. in [11] has shown that the blacklist is less effective against zero-hour phishing.

The image-based approach is another type of antiphishing solution that is based on the analysis of website images. Most of the time, this approach will store information of the website in a database. The information includes the visual layout and the captured images. Then, the information from the query website is compared with the database to determine the level of similarity. Image-based approaches have received considerable attention because of their ability to overcome the limitations imposed by the text-based antiphishing approaches [12, 13]. This approach includes utilizing the image processing tools (e.g., ImgSeek [14] and OCR technology [15]) and image

processing techniques (e.g., SIFT [16]). This approach is very effective against phishing websites that target legitimate websites whose information is already in the database. However, to be effective, this approach requires a comprehensive database. In other words, it may cause false alarms to legitimate websites that have not been registered in the database. The effectiveness of this approach also depends on the quality of the image. For example, OCR may extract incorrect data if the image is blurry. In addition, antiphishing solutions based on visual layout will fail if the phishing website contains dynamic content such as advertisements.

Another type of antiphishing solution is to utilize the search engine. This type of antiphishing solution will leverage the power of a search engine and perform further analysis based on the returned search results to determine the legitimacy of a website. Usually, this solution uses popular search engines like Google, Yahoo, and Bing. The results returned by these search engines are very sensitive to the search query. Each entry in the search results is usually listed in order of importance related to the search query. There are many studies that use search engines to check the legitimacy of a website. For example, Naga Venkata Sunil and Sardana [17] proposed a method that uses website ranking derived from the PageRank [18] to examine the legitimacy of the website. Huh and Kim [19] also proposed a similar method that uses website ranking to distinguish legitimate websites from phishing websites. Instead of using PageRank to check the ranking of a website, they compare the rates of growth in the ranking for new legitimate websites and new phishing websites. Similar methods can be found in [3, 4, 20]. This solution is effective against phishing websites because it is very rare for the search engines to include phishing entries in the search results. However, new legitimate websites may suffer because these websites usually do not have a high ranking in the search engine index.

3. Proposed Methodology

Typically, phishers like to imitate a legitimate website when developing phishing websites. They seldom make major changes to the content of phishing websites other than the inputs used to obtain personal credentials from potential victims. This is because major changes to the content of the website will only arouse user suspicion and increased workload. Besides, phishing websites usually have a short lifespan. The appearance of different phishing websites that are targeting the same legitimate websites will look similar to each other [14]. This appearance includes textual and graphic elements such as the favicon. Because the favicon is a representative of the website, this motivates us to use the favicon to detect phishing websites.

3.1. Website Favicon. The favicon is a shortcut icon attached to the URL that is displayed on the desktop browser's address bar or browser tab or next to the website name in a browser's bookmark list. Figure 2 shows an example of the Internet Explorer browser showing the PayPal favicon. The favicon represents the identity of a website in a 16 × 16 pixels' image

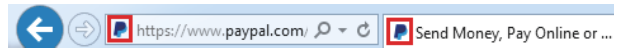

Figure 2: Example of PayPal's favicon displayed on the browser's address bar and tab.

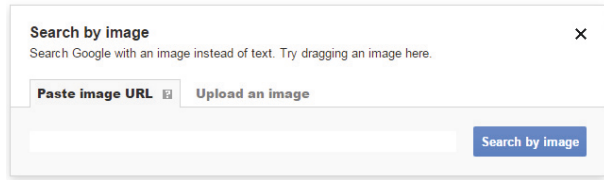

Figure 3: Example of a GSI interface.

file. It is also available in several different image sizes, such as 32 × 32, 48 × 48, or 64 × 64 pixels in size.

In order to access the favicon, we append the favicon.ico string to a website domain name. For example, given the PayPal website URL, https://www.paypal.com/, we extract the domain name (i.e., paypal.com) and append the favicon.ico to the end of the domain name, so that it becomes paypal.com/favicon.ico. The newly formed URL will be fed into the Google search by image engine to obtain information related to the favicon.

3.2. Google Search by Image. By default, the Google search engine allows text to be used as a search query to look up all sorts of images. In addition, Google also allows Internet users to search for information based on the content of an image. This mechanism of search by image content is essential for our proposed method to retrieve the correct information about an image. Figure 3 shows an example of the Google search by image (GSI) interface. Basically, there are two options to use GSI:

(i) Paste image URL: this option allows the user to directly use the image's URL found on the Internet.

(ii) Upload an image: this option allows the user to use images from local drives of computers. In addition, it also allows users to drag and drop images directly into the interface.

The Google search by image is an image query function with a Content-Based Image Retrieval (CBIR) approach and it returns a list of information specific to the query image. It extracts and analyzes the content (i.e., colours, shapes, textures, etc.) of the query image to find matching image data from the search engine database. The main difference between the search by image and normal image search is that the search by image utilizes image content to find matching image data while the normal image search uses metadata such as keywords, tags, or descriptions associated with the image to find matching image data. Figure 4 shows an example of the search results returned by GSI when the PayPal favicon is queried. We employ the *Paste image URL* option into our proposed method to feed the favicon into the GSI. To achieve this, we use a custom Application Program Interface (API) developed by Schaback [21], as shown in Figure 5. This API

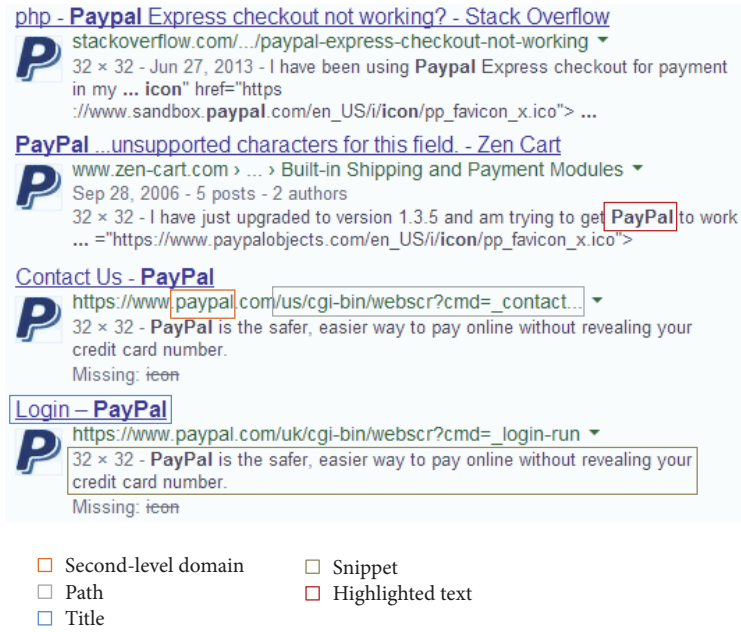

Figure 4: Example of GSI results when the PayPal favicon is queried.

```
https://www.google.com/searchbyimage?&image_url=<url>
```

Figure 5: Snippet code of GSI API.

utilizes the GSI to return a list of search entries relevant to the query image.

3.3. Proposed Features. We have a total of five features for this work. Four are extracted from the search result (refer to the red boxes outlined in Figure 4). They are as follows:

(i) Second-level domain (SLD): the SLD is the name by which a website is known. It is located directly next to the top-level domain (TLD). For example, in http://www.mydomain.com/, *mydomain* is the SLD of *.com* TLD. We propose using the SLD as part of our features because it has a high possibility of revealing the identity of a website based on the search results. Thus, the SLD is extracted from a list of entries returned by the GSI. In order to avoid confusion towards the counting, we use the term "unique term" to represent each unique SLD extracted from the entries. Therefore, the frequency of each unique term is computed from the number of occurrences of SLDs found from the search result.

(ii) Path in URL (path): this path is usually located after the top-level domain of a URL. For example, in https://www.domain.com/image/index.php, */image/index.php* refers to a unique location for a file named *index.php*. The path is used as part of our proposed features because the search results often contain terms in the path that are associated with the identity of the targeted legitimate website. While phishers can change the path of a URL in a different way, they must maintain the identity keyword in the URL in order to convince Internet users that they are visiting the correct destination. For this reason, we extract the full path of each URL from the search results returned by GSI. Then, we use the unique terms extracted in advance to find matching identities from all paths. Hence, in order to capture this property, the number of occurrences for every unique term found in the path is recorded.

(iii) Title and snippet (TNS): the title is the text that appears on top of the URL, and the snippet is the description that appears below the URL. We observed that the identity of the website does not always appear in the URL of the search entries. Instead, the identity of the website can also be found in the title or snippet of the search entries. Therefore, each unique term extracted beforehand is used to find a match in all the titles and snippets of search results. Hence, the number of occurrences for each unique term found in the titles and snippets is recorded.

(iv) Highlighted text (HLT): highlighted text is the bold text that appears in the title or snippet of an entry in the search results. The highlighted text indicates the most relevant and important keyword for the search favicon. This feature is very important and has a high tendency to reveal the true identity of the favicon. This is due to the fact that the content of a favicon must comply with the image content defined by GSI in order to have bold text appearing in the search results. Therefore, the proposed method extracts the bold text from each entry in the search results to find a match based on the unique terms extracted beforehand. The numbers of occurrence for each unique term found in all the highlighted text are recorded.

Table 1: Frequency of each unique term for each feature based on the entries in Figure 4.

Unique term	SLD	path	TNS	HLT
Stackoverflow	1	0	0	0
Zen-cart	1	0	0	0
PayPal	2	1	10	9
Total frequency count	4	1	10	9

Table 2: Weight assigned to the proposed features.

Feature	Notation	Weight
SLD	w_{sld}	20
Path	w_{path}	10
TNS	w_{tns}	30
HLT	w_{hlt}	40

To demonstrate the formation of each feature, we use the example shown in Figure 4 and show the frequency count of each unique term in Table 1.

We use the following equations to calculate the weighted frequency for each unique term across each feature (i.e., SLD, path, TNS, and HLT).

$$\begin{aligned} h_i^{\text{sld}} &= \frac{f_i^{\text{sld}} * w_{\text{sld}}}{F_{\text{sld}}}, \\ h_i^{\text{path}} &= \frac{f_i^{\text{path}} * w_{\text{path}}}{F_{\text{path}}}, \\ h_i^{\text{tns}} &= \frac{f_i^{\text{tns}} * w_{\text{tns}}}{F_{\text{tns}}}, \\ h_i^{\text{hlt}} &= \frac{f_i^{\text{hlt}} * w_{\text{hlt}}}{F_{\text{hlt}}}, \end{aligned} \tag{1}$$

h_i^{sld}, h_i^{path}, h_i^{tns}, and h_i^{hlt} refer to the weighted frequency of ith unique term for the four features. f_i^{sld}, f_i^{path}, f_i^{tns}, and f_i^{hlt} are the frequency count of ith unique term in which i is from the list of unique terms. F_{sld}, F_{path}, F_{tns}, and F_{hlt} are the total frequency count of all the unique terms under one feature. $w_{\text{sld}}, w_{\text{path}}, w_{\text{tns}}$, and w_{hlt} are the weight assigned for each feature (as shown in Table 2) and they are determined empirically. We assign the fourth feature the highest weight because the highlighted text usually has a high tendency to reveal the identity of the favicon. Based on preliminary experiments, we observed that the fourth feature has a very low frequency in the search results. It only occurs when the query favicon is truly matched with the image content stored in the Google image database. The first and third features are assigned with modest weight mainly because the frequency of identity depends on the Google search engine index. In other words, a false identity with a higher frequency can take over the real identity when the GIS cannot return enough information related to the favicon. We argue that these features are slightly lower in the level of importance compared with the fourth feature. The second feature is assigned with the lowest weight because phishers can always make changes to the path in a web address without affecting the whole website. In addition, we do not want Phishdentity to have a great effect if phishers exploit the path in a web address to avoid detection.

After obtaining the weighted frequency for each feature, we need to combine them to form the final frequency for each unique term. To do so, we use the following equation:

$$H_i = \frac{h_i^{\text{sld}} + h_i^{\text{path}} + h_i^{\text{tns}} + h_i^{\text{hlt}}}{w_{\text{sld}} + w_{\text{path}} + w_{\text{tns}} + w_{\text{hlt}}}. \tag{2}$$

3.3.1. Domain Name Amplification (DNA). A domain name is a unique name that is registered under the Domain Name System (DNS). It is used to identify an Internet resource such as a website. Typically, a legitimate website has a unique name differing from other websites. On the contrary, phishers are more likely to incorporate the name of a legitimate identity keyword into phishing URLs to confuse Internet users. Thus, this leads to the addition of the fifth feature, which is the DNA to the previous work.

Once we have computed the final frequency for all the unique terms, we need to amplify the final frequency of a unique term that corresponds to the SLD of a query website. To achieve this, we search through the list of unique terms to see if there is a match between the unique term and the SLD of the query website. If there is a match, we will increase the final frequency of the corresponding unique term by five percent.

$$H_i' = \begin{cases} 1.05 \times H_i, & \text{if } i\text{th unique term} = \text{SLD}_{\text{query}} \\ H_i, & \text{otherwise.} \end{cases} \tag{3}$$

Otherwise, we will append the SLD into the list of unique terms and count the frequency of all the terms for the three features (i.e., path, TNS, and HLT). This step is important because it can improve the detection performance. The rationale is that the GSI may not always include the entry for new and unpopular legitimate websites in the search results if they are not in the search index. Instead, the identity can be obtained from the other entries in the search results if the SLD of the query website is present in the list of unique terms. This is because the other entries might contain identity keywords in the path, TNS, or HLT. Next, we use the following equation to obtain a unique term with the highest final frequency:

$$\text{UT}_{\max} = \arg\max_i \left\{H_i'\right\}, \quad \text{where } i = 1, 2, 3, \ldots, n. \tag{4}$$

The unique term with the highest final frequency is deemed as the identity of the query website. If the identity of a query website does not match its SLD, we will assign a value of 1. Otherwise, we will assign a value of zero as below:

$$S_1 = \begin{cases} 1, & \text{if } \text{UT}_{\max} \neq \text{SLD}_{\text{query}} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $\text{SLD}_{\text{query}}$ is the SLD of a query website.

3.4. Additional Features. We are aware that there may be some websites that do not have a favicon. Phishdentity will become suboptimal if the favicon is missing from the website. Therefore, we have incorporated additional features, which are based on the URL, to compensate for the missing favicon. While phishers can conceal malicious content on the website, they cannot hide the URL or the IP address. There are many studies conducted to detect phishing websites based on the URL [19, 20, 22]. These studies produce fairly good results in the classification. For this reason, we have adopted five additional features based on the URL to the proposed method for classifying websites. To achieve this, we will extract the URL from a query website. Then, we apply feature extraction on the URL to obtain the required features. The proposed additional features are as follows.:

(i) *Suspicious URL*. This feature examines the URL for at-sign (@) or dash (-) symbol. If the at-sign symbol is present in the URL, it forces the string to the left to be removed while the string to the right is considered to be the actual URL. Thus, the user will be redirected to access the URL that is located to the right of the at-sign symbol. We note that some of the latest browsers (i.e., Google Chrome) still experience this problem when the at-sign symbol is used in the URL. Another reason is that there are many Internet users still using legacy browsers and this makes them vulnerable to this type of phishing attack. Phishers use this technique to trick Internet users who rarely check the website URL when browsing the Internet. Likewise, the dash is also often used in phishing URLs. Phishers imitate legitimate domain names by inserting dashes into the URL to make an unsuspecting user believes it is the legitimate domain name. For example, the phishing domain, http://www.pay-pal.com/ is imitating PayPal domain name, https://www.paypal.com/. However, the use of dashes in a domain name is rarely seen on a legitimate website. This technique can easily deceive users who do not understand the syntax of the URL and cannot tell the difference in domain names. Thus, we will look for these symbols to identify suspicious URLs. In order to do that, we will tokenize the URL based on two delimited symbols (i.e., dot and slash). Next, we will find the at-sign and dash symbols by going through each token. If there is a matching symbol in the token, then we assign one for this feature. Otherwise, we assign zero.

(ii) *Dots in Domain*. During data collection, we note that it is very unlikely for a legitimate website to have more than five dots in the URL domain while most phishing websites have five or more dots in the URL domain. We reason that phishers use such tricks to obfuscate Internet users from perceiving the actual phishing URL. This feature is also mentioned in [4, 17]. Hence, we have adopted this feature in our proposed method to classify websites. In order to do that, we will count the number of dots that are present in a URL domain. We assign one for this feature if the domain has five or more dots. Otherwise, we assign zero for this feature.

(iii) *Age of Domain*. This feature examines the age of domain with the WHOIS service. Based on the experiments conducted, we observed that many phishing websites have a very short lifespan. Typically, they last from a few hours to a few days before disappearing from the Internet. CANTINA [4] proposed a similar feature to check the age of a website domain, but it is different than ours. Instead of using 12 months as the threshold to determine the legitimacy of a website, we proposed using 30 days to evaluate the query website. This is because there are a lot of new legitimate websites whose lifespan is less than 12 months. It is undeniable that there are some phishing websites that last longer than a week. However, the longest life expectancy ever recorded for phishing websites was 31 days according to the report published in APWG [23]. In addition, the report also showed that most of the phishing websites only have an average lifespan of a week. We assign one to this feature if the age of the query website is equal to or less than 30 days. Otherwise, we assign zero to this feature.

(iv) *IP Address*. The IP address is the numerical number separated by periods given by the computer to communicate with other devices via the Internet. During the data collection phase, we observed that there are some phishing websites using IP addresses in the URL (domain and path). However, we did not find any IP addresses on legitimate websites. It is very rare for legitimate websites to use an IP address as a website address for public access. This is because the IP address has no meaning apart from being an Internet resource identifier. Using an IP address is a cheap way to create a website because it can be done by using a personal computer as a web server. Therefore, phishers do not need to register a website address with any domain name registrar. For this reason, we extract the URL from the query website and look for the existence of an IP address. If the URL contains an IP address, we assign one to this feature. On the contrary, if the URL does not contain an IP address, this feature is assigned zero.

(v) *Web of Trust (WOT)*. WOT [24] is a website that displays the reputation of another website based on the feedback received from Internet users and information from third-party sources such as PhishTank and TRUSTe [25]. The Web of Trust has an API that can be used to inspect a website for its legitimacy. Figure 6 shows the WOT API snippet code used to retrieve the reputation of a query website. The *value* in the code is where we insert the query website, and the *api_key* is where we insert the API registration key to activate the API. Once the API has generated a reputation value for the query website, we compare the value based on the scale used by the WOT to evaluate the website. The reputation value used by the WOT is shown in Table 3 where a value of 80 and

http://api.mywot.com/version/interface?hosts=value&callback=process&key=api_key

Figure 6: Snippet code of the WOT API.

Table 3: Reputation rating of WOT.

Description	Very poor	Poor	Unsatisfactory	Good	Excellent
Reputation value	≥0	≥20	≥40	≥60	≥80

higher indicates that the website receives very good feedback from Internet users while a value of 19 and below indicates the website could endanger Internet users. To this end, this feature is assigned one if the reputation is rated less than 20. On the other hand, if the reputation of the query website is rated 20 and above, then this feature is assigned zero.

We use the following equation to formulate the calculation of additional features:

$$S_2 = \sum \left(w_j * h_j \right), \tag{6}$$

where S_2 is the score for additional features of a query website. h_j is the value obtained from jth additional feature. w_j refers to the weight assigned for each feature and it is determined empirically as shown in Table 4. The WOT feature is assigned with the highest weight mainly because it can display a website ranking. The ranking order can change based on votes received from the public and from the information obtained from third parties. The age of domain feature is assigned as the second highest weight because all website owners must register their websites with a hosting provider to obtain a meaningful domain name. Thus, phishers cannot easily fake the age of the website. We give a higher weight to the WOT than age of domain because it uses active information like the users' feedback and third-party listings to validate the legitimacy of the website. Suspicious URLs, dots in the domain, and the IP address are assigned the same weight. Mainly, they are the local features of the URL, and the data is not verified by any third parties. Therefore, we propose that the weight distribution of suspicious URLs, dots in domain, and IP address are a little lower than the WOT and age of domain.

3.5. Final Integrated Phishing Detection Scheme. To determine the legitimacy of a website, we use (7) to calculate the score. More precisely, we use scores derived from (5) and (6) as input to this equation. The score produced by this equation will be the final score for the website.

$$\text{FinalScore} = S_1 C_1 + S_2 C_2. \tag{7}$$

S_1 is the score obtained from (5) and S_2 is the score obtained from (6). C_1 is the weight given to S_1, while C_2 is the weight given to S_2. C_1 and C_2 are the optimum weights obtained from the experiments conducted in Section 4.2. The *FinalScore* is used to determine the legitimacy of the query website. If the *FinalScore* exceeds the threshold τ, then the query website is classified as phishing. Otherwise, the query website is classified as legitimate. Similarly, we have dedicated Section 4.2 to experiment with the different variants of this threshold. This experiment will provide the optimum threshold.

4. Experiments and Evaluation

We have implemented a prototype of Phishdentity. It is written in C# language using the Microsoft Visual Studio 2010 Professional Edition. To verify the effectiveness of Phishdentity, we have collected 5,000 phishing websites and 5,000 legitimate websites, from December 20 to December 26, 2017. The phishing websites are obtained from the PhishTank archive. In particular, we are only interested in collecting phishing websites that have not been verified and are still online during this period. For legitimate websites, we chose Alexa [26] to collect our data. We refined Alexa to return only the top 500 sites on the web by category. Categories include but are not limited to arts, business, health, recreation, shopping, and sports. During data collection, we found that there is a total of 165 websites (3.30%) from Alexa that did not have the presence of favicon. We also found that there is a total of 81 websites (1.62%) from PhishTank that did not have the presence of favicon. Figure 7 summarizes the distribution of the dataset.

We conducted three experiments to verify the effectiveness of the proposed method. Experiment 1 is designed to evaluate the detection performance of the proposed method without the additional features. Experiment 2 is conducted to assess the integration of the proposed method with additional features to classify websites with a missing favicon. Experiment 3 is conducted to benchmark the performance of the proposed method with other phishing detection methods. In experiments 1 and 2, we will use a total of 7,000 websites (3,500 legitimate and 3,500 phishing) for training and optimum parameters setup. In experiment 3, we use a total of 3,000 websites (1,500 legitimate and 1,500 phishing) to validate the proposed method. It is noteworthy to mention that the 3000 websites used in experiment 3 are a new set of samples that have not been used in experiments 1 and 2. The experimental results are displayed using the following measurement metric:

(i) True positive (TP): a phishing website is correctly classified as phishing.

(ii) True negative (TN): a legitimate website is correctly classified as legitimate.

(iii) False positive (FP): a legitimate website is misclassified as phishing.

(iv) False negative (FN): a phishing website is misclassified as legitimate.

(v) F-score (F_1): this shows the overall classification accuracy of the model and the equation was composed of $F_1 = 2\text{TP}/(2\text{TP} + \text{FP} + \text{FN})$.

Table 4: Weight for the additional features.

Feature, h	Suspicious URL	Dots in domain	Age of domain	IP address	WOT
Weight, w	0.1	0.1	0.3	0.1	0.4

Table 5: Phishdentity assessment results.

Test bed	TP (%)	TN (%)	FP (%)	FN (%)	F_1
(1) Phishdentity (DNA not included)	94.57	89.31	10.69	5.43	0.92147
(2) Phishdentity (DNA included)	97.00	95.54	4.46	3.00	0.96297

4.1. Experiment One: Evaluation of Phishdentity. Experiment one is designed to evaluate the performance of Phishdentity to classify the websites based on search results returned by Google. We used the default parameters specified in the GSI API to return the search results. We have designed two test beds for this experiment: (1) The first test bed is designed to classify the websites using SLD, path, TNS, and HLT. (2) We integrated a DNA feature in addition to all the features used in the first test bed to form the second test bed to amplify a unique term that corresponds to the SLD of query website.

Table 5 illustrates that the second test bed has shown improvement after integrating DNA into the test. More specifically, the use of DNA has increased the effectiveness of Phishdentity to find the identity of a website. In other words, the second test bed is able to locate the correct identity even if the search results may contain many false identities. This improvement is noticeable in classifying legitimate websites where there is a reduction of 6.23% in false positive numbers. However, the number of false positives is still considered high. There are a few factors that contribute to the high number of false positives. First, there are a total of 91 out of 3500 legitimate websites without the presence of a favicon. Therefore, this contributes a total of 2.6% to the false positive numbers. Second, the API can return incorrect information when feeding with a wrong version of a favicon. This happens when the query website has a newer version of the favicon than the one stored in the Google image database. Although this issue has contributed some false positives, we foresee that the new favicon will be soon available in the Google image database. It is also in accordance with the frequent update of the Google web crawler to the database. Third, there are 13 websites that have restricted the access to their favicons when we perform the test. On the other hand, the second test bed has shown some improvement in detecting phishing websites. It reduces 2.43% from 5.43% to 3.00% of false negatives. The decline in the number of false negatives is due to Google's ability to filter malicious entries from the search results. But it will take some time for Google to determine the legitimacy of new malicious entries because most phishing incidents usually occur in the early stage of the attack. We adopted the second test bed setup for subsequent experiments.

4.2. Experiment Two: Evaluation of Phishdentity with the Absence of Favicons. This experiment is designed to examine the performance of Phishdentity with the absence of a favicon. This experiment is important to our proposed method because it reveals the potential of Phishdentity in classifying websites. For this reason, we designed three series of experiments for this section. The first series will be used to determine the optimal weight of C_1 and C_2 in (7) using a threshold of 50 as the baseline. We recorded the number of correctly classified websites each time C_1 is increased by one and C_2 is reduced by one. We stopped the process when it produced the highest number of correctly classified websites. Once C_1 and C_2 have been determined, we used them to conduct the second series of experiments. The second series is designed to determine the optimal threshold, τ. To do so, we set τ to zero initially. Then, we observed the number of correctly classified websites each time τ is increased by one. We halted this process when it produced the highest number of correctly classified websites. The third series is used to demonstrate the performance difference with and without the integration of additional features into Phishdentity. Two test beds are designed for the third series. The first test bed was used to demonstrate the performance of Phishdentity without additional features. The second test bed integrated Phishdentity with additional features using τ obtained from the second series of experiments. Experiment two produced the final version of the Phishdentity and it was used in the next experiment.

Table 6: To find the optimum weight for C_1 and C_2 when τ is set to 50 as the baseline.

C_1	C_2	TP (%)	TN (%)	FP (%)	FN (%)	F_1
0	100	92.34	91.71	8.29	7.66	0.92050
20	80	94.77	93.20	6.80	5.23	0.94032
40	*60*	*97.14*	*95.89*	*4.11*	*2.86*	*0.96537*
60	40	97.0	95.54	4.46	3.0	0.96297
80	20	97.0	95.54	4.46	3.0	0.96297
100	0	97.0	95.54	4.46	3.0	0.96297

We observed that our Phishdentity with additional features achieved the lowest error rates (FP and FN) in classification when we allocated a weight of 40 to C_1 and a weight of 60 to C_2, as shown in Table 6. In addition, we were able to further reduce the error rate in the classification when τ is set to 60 using optimal weights obtained from the first series of experiments, as shown in Table 7. Based on our preliminary observations, the additional features can improve the weaknesses of a missing website favicon, as shown by the second test bed in Table 8. This reduced the false positive by 0.57%. However, the solution is subjected to a higher false negative where it increases the number of false negatives from 3.00%

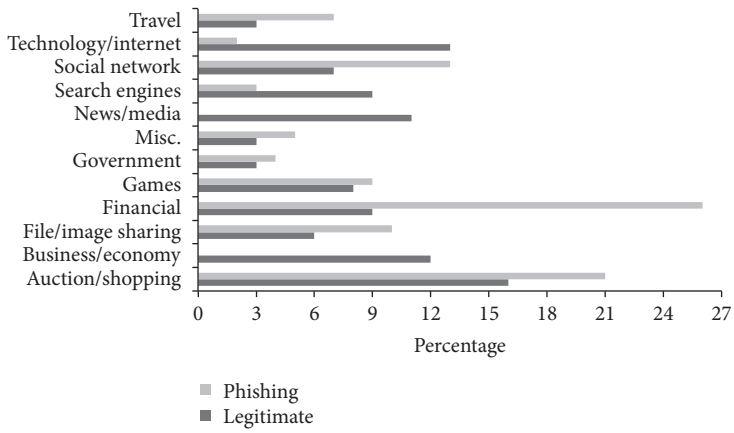

Figure 7: Categorization of legitimate and phishing websites.

Table 7: To find the optimum threshold, τ, for Phishdentity with additional features.

Optimal threshold, τ	TP (%)	TN (%)	FP (%)	FN (%)	F_1
0	99.80	90.23	9.77	0.20	0.95243
20	98.86	91.66	8.34	1.14	0.95425
40	97.34	94.40	5.60	2.66	0.95930
50	97.14	95.89	4.11	2.86	0.96537
60	*96.97*	*96.11*	*3.89*	*3.03*	*0.96555*
80	65.17	98.46	1.54	34.83	0.78184
100	23.34	100	0.00	76.66	0.37847

Table 8: Performance comparison for Phishdentity with and without additional features.

Test bed of Phishdentity	TP (%)	TN (%)	FP (%)	FN (%)	F_1
(1) Without additional features	97.00	95.54	4.46	3.00	0.96297
(2) With additional features	96.97	96.11	3.89	3.03	0.96555

Table 9: The score for each feature in Phishdentity.

Feature	Score
Favicon	40
Suspicious URL	6
Dots in domain	6
Age of domain	18
IP address	6
WOT	24
Total	100

to 3.03%. We argue that an increase of 0.03% in false negative is acceptable, given that the number of false positives was reduced by 0.57%. This showed that a solution based on the URL can be used to improve the detection results.

With the weight allocated to C_1 and C_2 as 40 and 60, respectively, the score for each feature is given in Table 9. Note that the weight of 60 for C_2 is a combination of scores from the suspicious URL, dots in domain, age of domain, IP address, and WOT feature. These scores are obtained and converted from the distribution shown in Table 4.

4.3. Experiment Three: Evaluation of Final Phishdentity. In order to show the effectiveness of Phishdentity in classifying websites, we perform benchmarking with other antiphishing methods. In this experiment, we chose CANTINA [4] and GoldPhish [15] for the benchmarking. CANTINA is a content-based antiphishing approach that utilizes the term frequency-inverse document frequency (TF-IDF) technique. It extracts five of the most important keywords from the website and feeds them to the Google search engine. Then, to determine the legitimacy of a website, it searches for a matching domain name from the search results. On the other hand, GoldPhish is an image-based antiphishing approach that utilizes the optical character recognition (OCR) technique. First, it uses a predefined resolution to capture a screenshot of the website. Then, OCR extracts the textual information from the captured screen and feeds this to the Google search engine. Next, it searches for a matching domain name from the search results to determine the legitimacy of a website.

Based on the experimental results in Table 10, CANTINA has falsely classified 5.87% of legitimate websites as phishing. We found that the TF-IDF does not work well for some of the legitimate websites. It will produce incorrect lexical signatures for the Google search engine if the query website

Table 10: Benchmarking results for final Phishdentity, CANTINA, and GoldPhish.

Antiphishing method	TP (%)	TN (%)	FP (%)	FN (%)	F_1
(1) Final Phishdentity	96.93	95.87	4.13	3.07	0.96419
(2) CANTINA	76.20	94.13	5.87	23.80	0.83704
(3) GoldPhish	98.07	80.13	19.87	1.93	0.89997

contains very little textual information that can describe the website. This issue is also discussed by Zhang et al. [4] where they planned to investigate for alternative approaches. From the evidence produced by this experiment, CANTINA does not work well for phishing websites. It falsely classified 23.8% of phishing websites as legitimate. We found that a relatively high number of misclassified phishing websites is due to the total weight assigned to additional features of the CANTINA. In other words, the total weight of the additional features can outweigh the TF-IDF in spite of the phishing websites that have been initially detected by the TF-IDF. Our proposed technique will not suffer this weakness because the final Phishdentity utilizes the favicon as the main input to GSI.

Table 10 shows that GoldPhish performs badly in classifying legitimate websites. The number of falsely classified legitimate websites is as high as 19.87%. We argue that GoldPhish faced several limitations when using the OCR tool to extract the textual information from the screenshot. First, GoldPhish is using a fixed size window to crop the screenshot. This will potentially exclude some important content that is located outside of the cropped region. Second, we noticed that most of the websites from Alexa have advertisements on the home page. There are some advertisements that are so large that they cover half of the screenshot. This has caused the OCR to extract incorrect messages. Third, some legitimate websites use many images on the homepage. This can cause the OCR to capture an image that does not contain important messages about the website. Fourth, the OCR does not work well with some of the uncommon typefaces and size of font. This makes the OCR unable to recognize the characters correctly. Among the three methods, GoldPhish performs the best in detecting phishing websites at 98.07%. This can be attributed to the ability of the Google search engine to return little or zero information about the phishing websites.

4.4. Limitations and Discussions. We discovered that our proposed technique has three limitations to the experiments. The first limitation is that phishers can slightly change the content of a favicon so that it is still familiar from the victim's point of view, but it does not retrieve legitimate websites. The phishers can also replace a favicon similar to another legitimate website. This can happen when Google has not yet crawled all the new updates for the parent company and its subsidiaries. However, this limitation is not vital. The GSI engine would extract different contents from the altered favicon and return information not related to the targeted legitimate website. In addition, phishing domains with altered favicons are less likely to appear in search results due to their relative young age. Thus, phishing websites with altered favicons can still trigger Phishdentity detection.

The second limitation is the false classification of the new or unpopular legitimate websites. New websites may not be indexed by Google and WOT may not have the information regarding these websites. Thus, the proposed method may falsely classify these websites as phishing. However, as time progresses, eventually the new websites will be listed in the WOT database. As for the unpopular websites, most likely they will not be targeted by the phishers. We believe that the missing data will be made available in the WOT database soon. This is because the WOT has a very large community which actively updates the information about the old and newly discovered websites. Furthermore, we have proposed an additional approach that is based on the website URL for classification. This additional approach is lightweight and can be used to offset Phishdentity's inability to classify the websites that do not have the presence of a favicon. Also, this additional approach will reduce the probability of such misclassification as evidenced by low false positive in experiment 3.

It is noteworthy to mention that the additional features are not performing well for some legitimate websites. First, the WOT categorizes pornographic websites listed in the Alexa ranked as dangerous websites. While this type of website has a lot of malicious advertising that will infect a computer with a virus, it is still classified as legitimate in the final calculation. The rationale is that we cannot deny that every legitimate website is harmless (i.e., pornographic websites), but our objective here is to determine the legitimacy and not the danger of the website. Second, WOT will give a low rating for e-commerce websites that do not use a secure connection for users to log in or make digital payments. The use of a secure connection on webpages can prevent eavesdropping, but it can be costly to implement, particularly for an e-commerce website targeting local businesses only. Nevertheless, this is unlikely to cause false alarms to legitimate websites because legitimacy is not solely determined by WOT.

5. Conclusion

In this paper, we have proposed a method known as Phishdentity. It is an approach that is based on the favicon to find the identity of a website. In order to retrieve information about the favicon, we use GSI API to return a list of websites that match the favicon. We have introduced simple mathematical equations to assist in retrieving the right identity from the many entries of search results. After that, we can reveal the identity of a website based on a unique term derived from the search results. We have also integrated additional features based on the URL to overcome Phishdentity's limitations. The additional features are used to address the scenario

where a favicon is missing from the website. In addition, we have conducted several experiments using a total of 10,000 websites obtained from Alexa and PhishTank. The experiments show that Phishdentity can achieve promising and reliable results.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The funding for this project is made possible through the research grant obtained from Universiti Malaysia Sarawak under the Special FRGS 2016 Cycle [Grant no. F08/SpFRGS/1533/2017] and Postdoctoral Research Scheme.

References

[1] APWG, http://docs.apwg.org/reports/apwg_trends_report_q2_2014.pdf.

[2] M. Cova, C. Kruegel, and G. Vigna, "There Is No Free Phish: An Analysis Of Free And Live Phishing Kits," *Proceedings of the Second USENIX Workshop on Offensive Technologies*, 2008.

[3] K. L. Chiew, E. H. Chang, S. N. Sze, and W. K. Tiong, "Utilisation of website logo for phishing detection," *Computers & Security*, vol. 54, pp. 16–26, 2015.

[4] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th International Conference World Wide Web (WWW '07)*, pp. 639–648, May 2007.

[5] J. C. S. Fatt, C. K. Leng, and S. S. Nah, "Phishdentity: Leverage website favicon to offset polymorphic phishing website," in *Proceedings of the 9th International Conference on Availability, Reliability and Security, ARES 2014*, pp. 114–119, Switzerland, September 2014.

[6] PhishTank, http://www.phishtank.com/.

[7] RSA, http://www.emc.com/collateral/fraud-report/online-fraud-report-1012.pdf.

[8] APAC, "Briefing on Handling of Phishing Websites in November 2016," http://en.apac.cn/Briefing_on_Handling_of_Phishing_Websites/201701/P020170112578956574716.pdf.

[9] "Malaysian Computer Emergency Response Team," http://www.mycert.org.my/en/services/advisories/mycert/2014/main/detail/955/index.html.

[10] S. Sheng, B. Magnien, and P. Kumaraguru, "Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish," in *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS '07)*, Pittsburgh, Pa, USA, July 2007.

[11] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *Proceedings of the 6th Conference on Email and Anti-Spam, CEAS 2009*, usa, July 2009.

[12] A. Herzberg and A. Jbara, "Security and identification indicators for browsers against spoofing and phishing attacks," *ACM Transactions on Internet Technology (TOIT)*, vol. 8, no. 4, pp. 16:1–16:36, 2008.

[13] Binational Working Group, https://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/archive-rprt-phshng/archive-rprt-phshng-eng.pdf.

[14] M. Hara, A. Yamada, and Y. Miyake, "Visual similarity-based phishing detection without victim site information," in *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security (CICS '09)*, pp. 30–36, IEEE, Nashville, Tenn, USA, April 2009.

[15] M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: using images for content-based phishing analysis," in *Proceedings of the 5th International Conference on Internet Monitoring and Protection (ICIMP '10)*, pp. 123–128, Barcelona, Spain, May 2010.

[16] S. Afroz and R. Greenstadt, "PhishZoo: detecting phishing websites by looking at them," in *Proceedings of the 5th Annual IEEE International Conference on Semantic Computing (ICSC '11)*, pp. 368–375, Palo Alto, Calif, USA, September 2011.

[17] A. Naga Venkata Sunil and A. Sardana, "A PageRank based detection technique for phishing web sites," in *Proceedings of the 2012 IEEE Symposium on Computers and Informatics, ISCI 2012*, pp. 58–63, Malaysia, March 2012.

[18] Open SEO Stats, http://pagerank.chromefans.org/.

[19] J. H. Huh and H. Kim, "Phishing Detection With Popular Search Engine: Simple And Effective," in *Proceedings of the in Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security*, pp. 194–207, 2011.

[20] R. B. Basnet and A. H. Sung, "Mining web to detect phishing URLs," in *Proceedings of the 11th IEEE International Conference on Machine Learning and Applications, ICMLA 2012*, pp. 568–573, USA, December 2012.

[21] A. Schaback, http://skyzerblogger.blogspot.tw/2013/01/google-reverse-image-search-scraping.html.

[22] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "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 (KDD '09)*, pp. 1245–1253, July 2009.

[23] APWG, http://docs.apwg.org/reports/APWG_Phishing_Attack_Report-Jul2004.pdf.

[24] "Web of Trust," https://www.mywot.com/wiki/api.

[25] "True Ultimate Standards Everywhere," https://www.truste.com.

[26] "Alexa," http://www.alexa.com/topsites.